FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY MALAYA

FAMILY INFORMATION SYSTEM
(FAMILYIS)

BY
Ang Seok Luan

Under the supervision of

Puan Norizan

Session 2001/2002

Dissertation submitted in partial fulfillment of the requirement for
Degree of Bachelor of Information Technology
University Malaya

ABSTRACT

Computer information system and computer applications are very important to all the organizations. Thus the government is introducing the MSC or Multimedia Super Corridor. It is meant to develop Malaysia into a regional and international technology and telecommunications hub in 2020. The MSC will propel the transfer of technology and become the test bed for R&D in high-tech industries.

Thus, Family Information System will be an example of what the function of family management in Smart Family will be like. The family management function is one most important feature in this system.

This report is about the development of Family Information System, **FamilyIS.** The system provides the facility to keep track of the usage of assets in a family. It includes an asset module that will inventories special equipments and apparatus in a family.

There are seven chapters in this proposal, which includes Introduction, Literature Review, System Analyst, System Design, System Implementation, Testing, and System Evaluation and Conclusion. The first four chapters are considered as phases that a system analysts need to go through before one can come out with clear and concise definitions of all the functional and non-functional requirements that are needed to develop a good application later in the future. The last three chapters are the remaining phases of system development, which uses all findings and information gather in the earlier stages.

ACKNOWLEDGEMENT

First of all, I would like to thank Puan Norizan, my supervisor for giving me this opportunity to develop this project. Secondly, I would like to thank her for her constructive advice, generous guidance, encouragement, support and supervision along the project. Her diligence and kindness in helping me throughout the project is deeply appreciated.

Thanks to Encik Ali Fauzi, as the project moderator who contributed suggestions and ideas to further enhance value of this project.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

Application – A business computer system that processes a portion of a database to meet a user's information needs. It consists of menus, forms, reports, queries, and an application program.

API – See Application Program Interface.

Application Program Interface – A set of program procedures or functions that can be called to invoke a set of services. The API includes the names of procedures and functions and a description of the name, purpose, and data type of parameters to be provided. For example, a DBMS product could provide a library of functions to call for database services. The names of procedures and their parameters constitute the API for that library.

Candidate Key – An attribute or group of attributes that identifies a unique row in a relation. One of the candidate keys is chosen to be the primary key.

CBIS - Computer Based Information System

Database - A collection of integrated records that are kept in files for future reference.

Hardware - The mechanical devices that comprise a computer system, such as the central processing unit, monitor, keyboard, and mouse, as well as other equipment like printers and speakers.

DBMS – Database Management System. A set of programs used to define, administer, and process the database and its application.

Foreign Key – An attribute that is the key of one or more relations other than the one in which it appear.

MIS - Management Information System

Normal Form – A rule or set of rules governing the allowed structure of relations. The rules apply to an attribute, functional dependencies, multivalued dependencies, domains, and constrains. The most important normal forms are 1NF, 2NF and 3NF.

Normalisation – The process of evaluating a relation to determine whether it is in a specific normal form and, if necessary, of converting it to relations in that specific normal form,

ODBC - Open Database Connectivity, a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data.

PB - Power Builder is a graphic PC-based client/server application development environment.

Primary Key – A candidate key selected to be the key of relation.

Search engine -A tool that enables users to locate information on the World Wide Web. Search engines use keywords entered by users to find Web sites, which contain the information sought. Some search engines are specifically designed to find Web sites intended for children.

FamilyIS – Family Information System

Software - A computer program, which provides the instructions, which enable the computer hardware to work. System software, such as Windows or Mac OS, operate the machine itself, and applications software, such as spreadsheet or word processing programs, provide specific functionality.

VB6 - Visual Basic 6.0 an event-driven programming language that are going to be used in developing SFMS.

4GL - 4th Generation Language

CHAPTER ONE

INTRODUCTION

CHAPTER 1 INTRODUCTION

1.1 PROJECT BACKGROUND

This project is about the development of application software designed to assist in management of the family's information. This application or system will be called Family Information System (FamilyIS). FamilyIS is specially designed to meet the requirements of the Family Members in managing the family's information, so that the information can be better managed and safeguard.

Strictly stated here is that FamilyIS is not designed to replace the talents of the Family's Members, but will provide all the power of the latest technology available to them in order to facilitate their work and reduce the repetitive work. A computer has never been able to replace an expert in any domain because the role of computers is just to provide a powerful management tool to all who needed it.

The system will have a database to keep all the information and data needed on family's information as well as family's property, family's account as well as the family's billing system. A database is now such an integral part of our daily life that often we are not aware that we are using one. FamilyIS will be aimed at developing a computerized database system where it will handle data entry, file maintenance and generation of a fixed set of specific reports. FamilyIS also include system administrator control that only allows user to view certain subsystem. It will collect and preserved information on facilities in which in the past has been available only on paper.

FamilyIS will be an example of what the function of family's information management in Family will be like. The Family's management function is one of the important aspect that must be given a serious consideration in accordance with the ever-changing family's lifestyle.

1.2 STATEMENT OF THE PROBLEM

The decision to build the application software to manage the family's information is indeed the consequences from the difficulties and problem arise from the manual filing system.

The manual filing system is too time consuming as well as it involves a lot of clerical work. A lot of family information are not probably used due to poor maintenance and the human inability to keep up with the increasing information in a particular family.

Therefore, a management information system is the best solution for all the difficulties and troubles faced. But as stated earlier the system is not meant to replace the Family. FamilyIS will also allow the user to keep information about particular billing information in the family, property of a family, and the bank account of a family.

1.3 OBJECTIVE OF THE PROJECT

The objectives for developing Scholl Facilities Management System are as followed:

i. To provide at the desktop all graphics and textual information needed for the management of family's information in a particular family

ii. To help in tracking the usage of fund in the family.

iii. Keeping the latest records of family's property, family's account and family's billing.

1.4 SCOPE AND LIMITATION OF FamilyIS

i. This project is a management information system that will be installed into one computer and the computer will act as the host of the whole system.

ii. The system can only be accessed locally (standalone) due to time limitation in developing the FamilyIS system.

iii. This system will only keep track of the Family's information, such as accounts, billing and property.

iv. For further enhancement of the project, in the future, the system is hoped to maintain the entire family's information system in a family, which include family's insurance and etc.

The project will be divided into modules as below:

Module 1: Method of access

The system is designed to allow authorized users to access FamilyIS by using a user ID and password. Only valid user is allowed to access into FamilyIS. Not every user has the same authorization to the system. The family's parents will have the full authorization in changing all the records in the database, so as the daughter's and son's who will be in charge of maintaining the family's information system.

Module 2: Tracking Usage

The system is designed to track all usage of every information in the particular family. User will be able to view possible information of a particular person's in the family. The

user also can obtain information on types of property stored in the family. Thus, parents can make decision whether to make use of the property for their usage.

Module 3: Maintenance

The system will give appropriate message regarding when will be the best time to do maintenance of needed assets. A pop up window will appear when the user run the system. This is to ensure that there will be a proper maintenance system for all the assets in the house.

Module 4: Assets

The system will keep track of all the fixed assets in the family, where it will keep information on the types of assets available, their quantity, their quality and their vendors. This can also serve as a system to store all the information of an asset in the event of any insurance claim.

Module 5: Report Generation

FamilyIS will be designed to generate report needed for the Family Members. The report can be printed ac-hoc or it can be set to print the require report on periodic basic.

1.5 PROJECT SCHEDULE

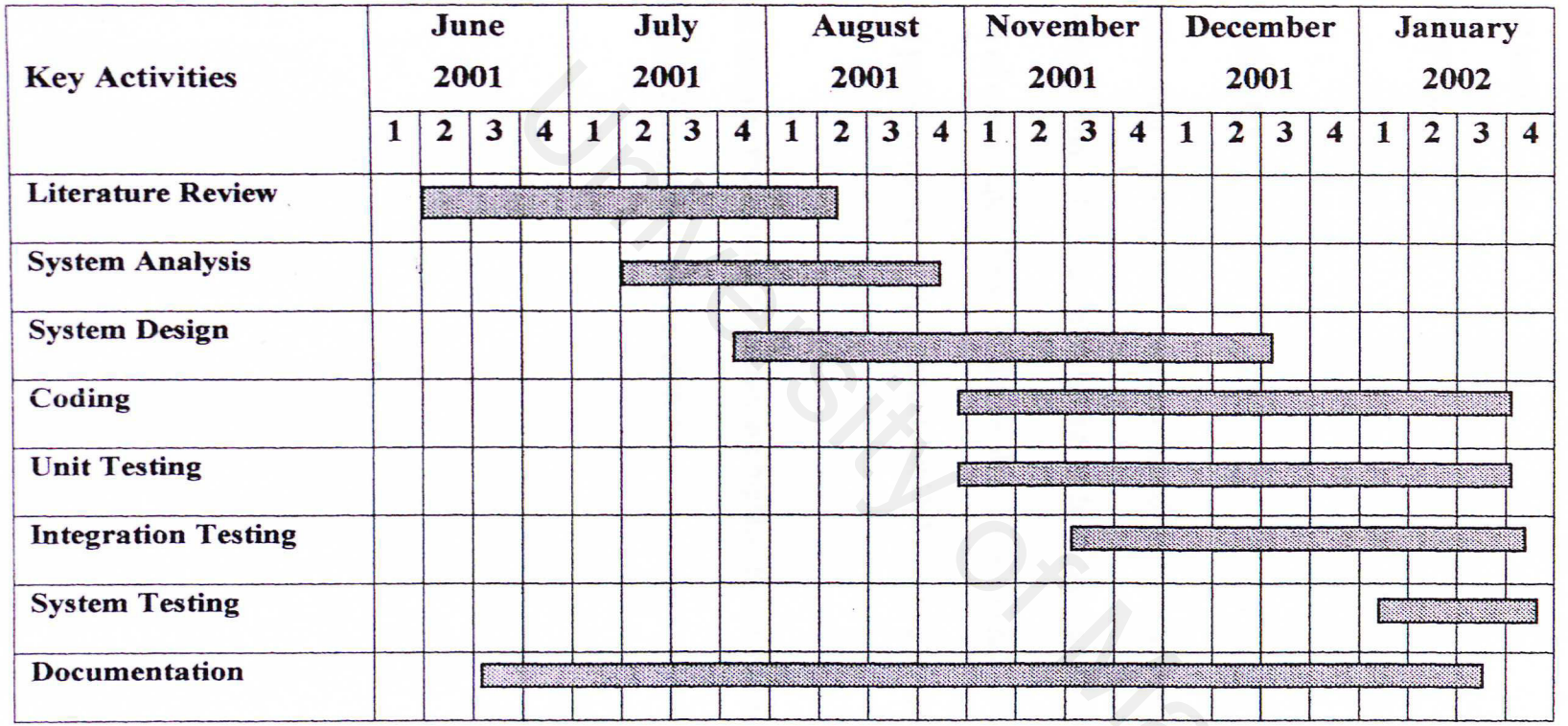


Figure 1.1 Gantt Chart For FamilyIS

1.6 RESEARCH PLAN

The research plan to gather information for developing FAMILYIS is as followed:

- Questionnaire
- Printed material such as previous thesis
- Internet surfing
- Brainstorming

1.7 SUMMARY

Family Information System is a computer based information system that manages the family's information. The system is going to help the Family members to track the information of every aspect listed in the database. It also enables the user to inventories the family's property as well as the accounts and billings in a particular family.

In the next chapter, the literature reviews are highlighted. It will show how other people develop application(s) that is similar to this one.

CHAPTER TWO

LITERATURE REVIEW

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION TO FAMILY INFORMATION SYSTEM

The Oxford International Dictionary of English defines families as group consisting of parents, children and close relative. In my case, family are restricted to those found in a typical family, especially family that consist of parents and children. Family typically include father, mother, sisters and brothers. Special apparatus and equipments such as projector, radio and television are also regarded as family facilities. These facilities will help to give a better environment for family to keep better record of the family matters. Nowadays, there are more and more matters and details that is needed to be sort out by a family as our world is changing to be more and more demanding. Due to increasing complexity in maintaining a family matters, computerized family information system is indeed a must so that the family details are not wasted and misuse.

The Family Information System will acts as the central data source for comprehensive and detailed information about personal details, billing system, equipment and assets available to the family. Family administrative person will identify the family components and assets that are needed to be manage. Information tracked about these elements in conjunction with built-in analysis and reporting then become the basis for good, sound family decisions.

2.2 FAMILY INFORMATION SYSTEM AS A MANAGEMENT INFORMATION SYSTEM

Family Information System is a Management Information System that can evolve into a Decision Support System that can assist the family administrative person in decision-making.

Family Information System is a computer-based system that generates timely and accurate information for the parents and children of a family. Family Information System is a MIS where it will help the family administrative person to manage and control the usage of family matters. It will collect and preserved information on family in the past has been available only on paper.

2.3 RESEARCH ON EXISTING PACKAGES

Family Origins System (www.familyorigins.com)

This particular system is developed to computerize the genealogical system in a family. The system includes the geographical location for each family member so that the family members could know where each member is located, so that they can easily access information about each member to provide a better understanding to family members. The system provides the capabilities to access, records and manage the entire family members in the database. It also includes the picture of each family member. However the system was developed for installation in a PC not for the network.

This system has good way of managing genealogical in family, but it should be installed in WAN, so that it can be access anywhere in the world, as most family member has a wide compound.

Web Address Book (www.webaddressbook.com) system is being developed to computerize the daily activities for family members. This system include the tasks that a family member would have do, a calendar and address book for all the family members to store their respective friends address. This system is set to provide a more efficient way for storing and managing the daily activities of all the family members because it can be access through the internet at any time given. This system suits the mobile lifestyle of today generation. In order to provide better accessibility and availability for the family members to their respective data, this system has cater for their needs by providing the capabilities to access, record and manage their data in the database. However, this system do not provide any reminder to the user in the event

of something happening, like, when the user has forget about his/her appointment at any given time, there is no reminder to let the user realized about the appointment. The reminder can be in the form of beeping sound or blinking light.

Faculty Information System (Azny 1996/1997)

The system was developed using PowerBuilder 5.0. with Window 95 interface. There are 3 functional modules, which was divided by its user. The modules are Students, Staff and Asset.

The Asset module is being analyst because it is more similar to Family Information System. In this particular module, it allows the user to do Asset Data Entry, New Item Purchase Detail, Malfunction Asset Info, Asset Viewing, View by Location, Complaint, and Report. The graphical user interface is relatively simple and easy to use.

2.4 TECHNOLOGY REVIEW

2.4.1 Visual Basic 6.0

Visual Basic is a simple, easy to learn language and programming environment, which can be used to build real application for window. It is widely used in business industry and IT industry for developing rapid prototypes of new applications. The features are show next:

Visual Basic's Technical Strength

a) Event-Driven Programming

An event is an action of some function, for example moving the mouse, selecting a item in a list, or clicking a button are events. These events drive what happens in Visual Basic program. This idea is quite different from traditional programming, in which we write lines of code that are executed sequentially. In event-driven programming, the code to run is determined by the type of event.

b) Visual Basic Control

Visual Basic is designed, in such a way that we can immediately design the windows that we wish. It has the ability to create and use self-contained components, or objects.

Control are elements you can use when designing a user-interface, just like the real life control. These controls can be used to display information or to take action.

Visual Basic control enables us to add features to our programs without having us involved in the details of simple drawing a control that accepts input.

c) **Data Access Objects (DAO) and ActiveX Data Objects (ADO)**

DAO can create complete, robust data management application. DAO act as a visual program internal representation of physical data stored in some type of database or data management engine. Think of Data Access Objects as special types of variables. These variables, however, represent data stored outside the program rather than information stored in the computer's memory while the programs running.

ActiveX Data Objects, or ADO, is most recent method of data access that Microsoft has introduced. ADO is intended to replace DAO, the original method of Visual Basic database access.

2.4.2 PowerBuilder

PowerBuilder is a graphic PC-based client/server application development environment. Using PB one can develop front-end applications which access RDBMs (Relational Database Management Systems) without coding in a 3GL(3rd Generation Language) such as C or C++. PowerBuilder uses it's own Powerscript, that is a basic-like language, that uses screens called painters to graphically put together applications. Powerscript is a 4GL (4th Generation Language).

PowerBuilder's Technical Strengths

a) **DataWindow**

The primary strength of PowerBuilder is its proprietary device called DataWindow. Some people have called it a product within a product. Others are still discovering new uses for it. It is the primary means by which a PowerBuilder application talks to the database. It has built-in features to format data for display,

allow different edit-styles, validate data entered by user, generate appropriate SQL based on the changes made by a user and also the RDBMS it is talking to and scores of other such invaluable features.

b) Object-Oriented (OO)

PowerBuilder is an object-oriented language. Though it is not a pure object-oriented language, it supports inheritance in most of the areas, permits encapsulation and enables polymorphism. Because of these reasons, it is possible to architect your applications in such a way as to reuse code within and across applications. If you make use of OO features, it also makes it simpler to maintain that application.

c) Native Drivers

Though ODBC (Open Database Connectivity) is good for accessing multiple databases through a common gateway, it covers only the common minimum features of these databases. PB provides native drivers for all the major RDBMSs, such as Oracle, Sybase, Informix, DB2, MS SQL Server...etc., so that you can take advantage of the power of these.

d) Cross Platform

You can write code once and run that application on all the flavors of Windows, namely Windows 3.1, Windows for WorkGroups, Win95 and Windows NT. You can also use the same code to run the application on Mac and Sun Solaris Unix.

e) **Web-enabled**

With PowerBuilder 5.0 and the Internet add-ons, you can build an application, which can access data in an RDBMS through a browser, whether it is on the corporate intranet or on the Internet.

Minimum configuration programming with PB

The minimum configuration recommended for PB is: 486 with 8 MB of RAM. The windows permanent swap file should be set to 12 MB. And don't have any other application running on the background.

2.4.3 ODBC

ODBC (Open Database Connectivity) provides a way for client programs (eg Visual Basic, Excel, Access, Q+E etc) to access a wide range of databases or data sources.

ODBC is a standardized API, developed according to the specifications of the SQL Access Group, which allows one to connect to SQL databases. It defines a set of function calls, error codes and data types that can be used to develop database independent applications.

ODBC is usually used when database independence or simultaneous access to different data sources is required.

2.4.4 Microsoft Access 2000

Microsoft Access 2000 is included in Microsoft Office 2000 pack. Using Microsoft Access, one can manage all the information from a single database file. Within the

file, data is divided into separate storage containers called tables. User can view, add and update table data by using online forms in Access. User also can find and retrieve just the data needed by using queries. They also can analyse or print data in a specific layout by using reports, a function in Access.

Microsoft Access 2000 is the Microsoft Office Database Management System available when one chooses to install the typical Microsoft Office

Access 2000 System Requirements

Here are the requirements to run Microsoft Access 2000:

- PC with a Pentium 75 megahertz (MHz) or higher processor
- Microsoft Windows® 95 or later operating system, or Microsoft Windows NT® Workstation operating system version 4.0 Service Pack 3 or later
- For Windows 95 or Windows 98: - 16egabytes (MB) of RAM for the operating system, plus an additional 8 MB of RAM for Access.
- 161 MB of available hard-disk space (Number indicates typical installation; your hard-disk usage will vary depending on configuration. Choices made during custom installation may require more or less hard-disk space.)
- CD-ROM drive
- VGA or higher-resolution monitor; Super VGA recommended
- Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device

Additional items or services required to use certain features:

- 9600 baud modem; 14,400 or higher-baud modem recommended
- Multimedia computer required to access sound and other multimedia effects
- Microsoft Outlook 2000 or Microsoft Outlook Express 5.0 or later required to run Office E-mail
- 8 MB of additional memory required to run Office E-mail
- Some Internet functionality may require Internet access and payment of a separate fee to a service provider

2.5 DEVELOPMENT METHODOLOGIES

The system development methodology is important during the development of software as it forms a common understanding of activities, researches, resources and constrains involved in a software development. When a methodology is identified, it can help us to find the inconsistencies, redundancies and omission in the process. There are many types of methodology available such as Waterfall Model, Prototyping, Transformation Model, Spiral Model and others. For my literature review, I have studied through the below methodology:

(1) Waterfall Model

One of the first models to be proposed is the waterfall model where stages are depicted as cascading form one another. One development stage will be completed before the next begins. Thus, when all the requirements are elicited from the user, analyzed for completeness and consistency, and documented in a requirement document, then the development team can go to the system design activities. The Waterfall model presents a very high level view of what goes on during development, and it suggests to developers the sequence of events they should expect to encounter.

The waterfall model can be very useful in helping developers lay out what they need to do. Its simplicity makes it easy to explain to users who are not familiar with software development.

During design, problems with requirements are identified, during coding, design problems are found and so on. The process is not a simple linear model but involves a sequence of iterations of the development activities.

Unfortunately, a model, which includes frequent iteration, makes it difficult to identify management checkpoint for planning and reporting. Therefore, after a small number of iteration, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won't do what the users wants. It may lead to badly structured systems as designed problems are circumvented by implementation tricks.

The problem with waterfall model is its inflexible partitioning of projects into these distinct stages. Delivered system is sometimes unusable, as they do not meet the uses real requirements. Nevertheless, the waterfall model reflects engineering practice. Consequently, it is likely that software process models based on this methodology will remain the norm for large hardware-software development.

(2) **Prototyping Model**

A prototype is a partially developed product that enables users and developers to examine some aspect of proposed system and decide if it is suitable or appropriate for the finished product. In other words, prototyping means building a small version of a system, usually with limited functionality that

can be used to help user or customer identify the key requirement of a system and demonstrate feasibility of a design or approach.

Prototyping is often used to design a good user interface: the part of the system with which the user interacts. Since the prototyping models allows all or part of a system to be constructed quickly to understand or clarify issues, it has the same objective as engineering prototype, where requirements or design require repeated investigation to ensure that the developer, user, and the customer have a common understanding both of what is needed and what is proposed. One or more of the loops for prototyping requirements, design or the system may be eliminated, depending on the goals of the prototyping. However, the overall goal remains the same, reducing risk and uncertainty in development.

Application prototyping, the process of developing and using the prototype, has five characteristics:

i) The prototype is a live, working application.

ii) The purpose of prototyping is to test out assumption made by analysts and users about required system features.

iii) Prototypes are created quickly.

iv) Prototypes evolved through an iterative process.

v) Prototypes are relatively in expensive to build.

Application prototyping has two primary uses. On the one hand, it is an effective device for clarifying user requirements. Written specifications are

typically created as a vehicle for describing application features and the requirements that must be met. Developing and actually using a prototype can be very effective way of identifying and clarifying the requirements an application must meet.

A second use of application prototyping is to verify the feasibility of a system design. Analyst can experiment with different application characteristics, evaluating user reaction and response. For instance, on method of interaction, whether through menus, special keys, or entry of keywords may be better than others for particular type of application than others. Processing procedures may change, leading to a more effective design. Creating a prototype and evaluating its designs through use will prove design feasibility or suggest the need to find other alternatives.

2.6 SUMMARY

After thorough studies of methodologies involved in this chapter, the proposed methodology chosen for this FamilyIS is Waterfall Model. Tools used to develop the FamilyIS must be able to support each other. The following are the tools used in the system:

i. Visual Basic 6.0

ii. ODBC (Open Database Connectivity)

iii. Relational Database

iv. Microsoft Access 2000

v. Structure Query Language (SQL)

There are not many available software and application available in the market that provides the function to track family records as well as asset management. Only few of the systems developed for the use in family. It can be said that none of the system review here are for the use in family in Malaysia.

The following chapter is where the analysis of the system needs is being done.

CHAPTER THREE

SYSTEM ANALYSIS

CHAPTER 3 SYSTEM ANALYSIS

3.1 INTRODUCTION

The main purpose of the system analysis phase is to learn exactly what takes place in the current system, and to determine and fully document in detail what should take place. The result of this process will be used to recommend improvement to the system. Through system analysis, the programmer may add, delete and modify system components toward the goal of improving the overall system. The information gathered during this phase has provided alternative strategies to develop this system. Through this phase also, the programmer can determine types of functional requirements and non-functional requirements for the system.

3.1.1 Objectives Of System Analysis

Following are some of the objectives of the analysis

i) To study the problem faced by the user

ii) To study the problem and find out the best solution to reduced it.

iii) To study how the new system will improve the current facilities management in a school.

iv) To acquire knowledge on how this system will be developed with the new emerging technology

v) Tools to develop the new system will be chosen among different types of new tools that have been studied and stated in chapter 2.

vi) To identify the major modules to be included in the system

vii) To identify what are the modules that are feasible to develop and the knowledge and tools need to have in order to develop them.

3.2 SYSTEM DEVELOPMENT METHODOLOGY

Before it is chosen, a methodology should reflect the goals of the development. Thus, after thorough studies of methodologies involved in Chapter 2, the proposed methodology chosen for this FamilyIS is Waterfall Model. Figure 3.1 shows the stages included in the model.

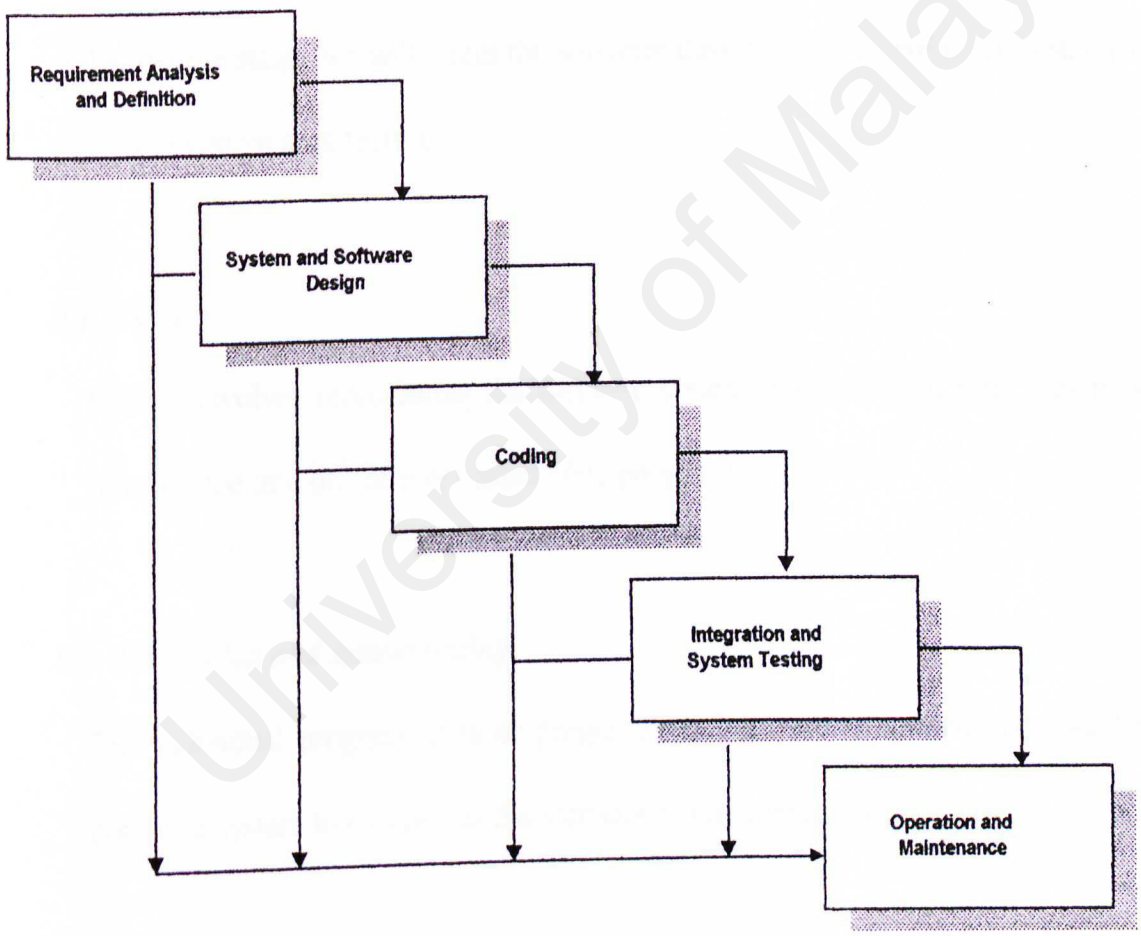


Figure 3.1 The Waterfall Model

The 5 stages of the Waterfall Model are discussed below:

i) Requirement Analysis and Definition

The concept, purpose and functionality of FamilyIS are identified and defined. During this stage, we have to study the existing system that is available in the market, and do planning for the new system. After finish this stage, it comes to system design.

ii) System and Software Design

Under this stage, we will begin the software design stage, where it will establish an overall system architecture.

iii) Coding

Coding involves representing the software system functions in a form that may be transformed into one or more executable programs.

iv) Integration and system testing

The individual program units or programs modules are integrated and tested as a complete system to ensure that the software requirements have been met.

v) Operation and maintenance

After testing, the system can be installed to its site to be fully utilized.

3.3 FACT FINDING TECHNIQUES

Facts finding is needed in order to have a better understanding of the system's needs and requirements. There are many sources that provide information in my research. The information gathering techniques involved are:

3.3.1 Study On Previous Projects

Researches are done by studying through some of the similar system done by other seniors in the faculty. Through this method, the programmer can collect data on how previous system were developed, what were the functional and non-functional requirements, and other related data.

3.3.2 Internet Research

Internet is used as the main resource for referring any ambiguities that arise during the entire development period. Through the Internet, the programmer not only collect some idea from the similar system, but they can also find out some interesting web design and feedback from the other remote users. Table 3.1 lists the keywords and search engine that I have used to gather information for this system:

KEYWORDS USED	SEARCH ENGINE
Family Management System	www.infoseek.com
Account Management System	www.altavista.com
Property Management System	www.yahoo.com
Billing System	www.britannica.com
Insurance System	www.ask.com
Sistem Pengurusan Kemudahan Keluarga	www.mol.com.my
Smart Family	www.catch.com.my

Table 3.1 Keywords and Search Engines

3.4 REQUIREMENTS SPECIFICATION

After completing the above fact-finding technique, the requirement is separated into three categories:

i) Requirements that absolutely must be met

a. Only the authenticate user is allow to use the system. Password and login ID need to be entered to gain access to the system.

b. The system should provide the functionality for the parents to add in new items or new information in the family together with its detail as well.

c. The system should let the parent to print out the report ac-hocly.

d. The system should let the Parents view the information according to different criteria.

ii) Requirements that are highly desirable but not necessary

a. Tool Tips should have for every control inside the system.

b. Help files should be provided in every window.

c. Error messages with some guidelines should be provided to the users when illegal operations happen.

iii) Requirements that are possible but could be eliminated

a. The style of the words for the report can be changed according to the users.

b. The user can change the graphical design easily.

3.5 FUNCTIONAL REQUIREMENTS

Following are some of the functional requirement of the system.

i) Log In

a. Before users can access into the system, the user need to provide login ID and password for verification

b. The main screen will be difference based on the user right for the users.

ii) Password Setting

This button let the users to change their password.

iii) Error Message

Display error message to guide users along the operation of the system.

iv) Data Manipulation

Allow user to manipulate the data entered earlier. For example to delete or update records.

v) View Summary and Report

This button lets the administrator to print out the daily and monthly report. The administrator can view other record based on different criteria as request.

3.6 NON-FUNCTIONAL REQUIREMENTS

Following are some of the non-functional requirement of the system:

i) Reliability

Reliability is the extents to which a system can be expected to perform its intended function with required precision and accuracy. Thus, the system should be reliable in performing its daily functions and operations. For example, whenever a button is clicked, the system should be able to perform some functionality or generate some message to inform the user what is happening.

ii) Scalability

The scalability is to promise the capability of the system to migrate as a client or server to machines of greater or lesser power, depending upon requirements, with little or no change to underlying components. Database scalability issues can be resolved using distributed database architecture whereas web application scaling can be addressed by increasing bandwidth or by additional web servers.

iii) Consistency

Refers to any similar screen design or interface requires or process a similar action.

iv) Usability

The system should be developed in such as way that it is easy to use. It will enhance and support rather than limit or restrict the office processes.

v) **Security**

The system should be equipped with sufficient security. Each access by the user should be authenticated and validated by the system. The system should not show any potential of leakage of information. The password should be encrypted.

vi) **Data and Services Backup**

The system should be able to restore to its normal operation from any potential disaster. There should be a second backup for data and services to ensure the continuous of operation.

vii) **On-Time**

The system should be developed within the given time frame. In this period of time, all the requirement and also testing should be completed.

viii) **Flexibility**

The system should have the capability to take advantage of new technologies and resources. The system should be able to implemented in the changing environment.

3.7 DEVELOPMENT ANALYSIS

An analysis was carried out on the development tools to find out the most suitable tools for the system. These tools include the entire platform, servers, development software and programming language. Besides considering the suitability of the tools to the requirement, the tools used must be able to support each other. The following are the tools used in the system:

3.7.1 Visual Basic 6.0

Visual Basic 6.0 (VB6) is chosen as the development tool in this project because of the following reason:

a) VB6 is one of the most popular programming tools in windows environment due to its RAD (Rapid Application Development) capabity that associated with it.

b) VB6 is embedded with search engine (JET engine 1.0) that comes from the family that is similar to the internal search engine (JET engine 2.0) of DBMS used. Example, Microsoft Access. Both JET engines share the same database format and hence, an intermediate conversion program such as ODBC is not required.

c) Furthermore, VB6 uses an event driven approach to program the system and is not a procedural language. An application developed with an event-driven model response to event that happens in the computer environments. Such events include the pressing of mouse button or call function from another application that running concurrently.

d) Lastly, most important here is, mostly schools in Malaysia use Microsoft products that support VB6.

3.7.2 ODBC

Abbreviation of Open Database Connectivity, a standard database access method developed by Microsoft Corporation. The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them. Since version 2.0, the standard supports SAG SQL.

3.7.3 Relational Database

Relational database is chosen because:

a) The data structure used by FamilyIS is relatively simple and thus, easy to be mapped into tables.

b) Since it is conceptually independent, no pointer or links are risible to the programmer, this ease the coding task.

c) The query language (SQL) is relatively simple and sufficient to implement FamilyIS system.

d) Queries may be expressed without the use of iteration or recursion. A block of data is directly return to DBMS.

Figure 3.2 shows the structure of a relational database.

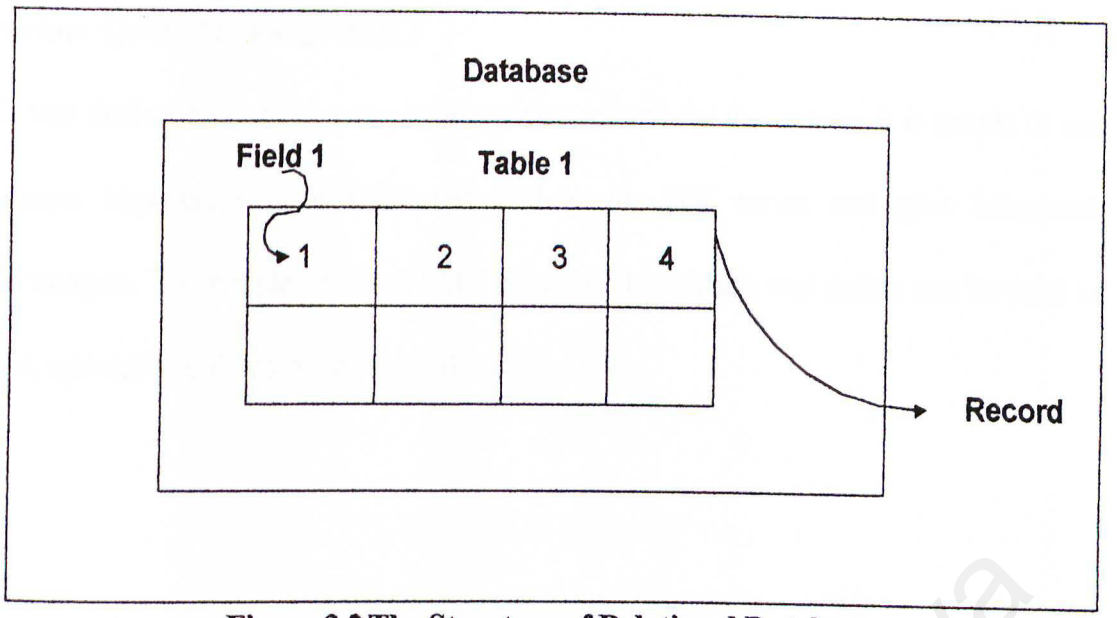


Figure 3.2 The Structure of Relational Database

3.7.4 Microsoft Access 2000

From the development technologies reviewed in Chapter 2, Microsoft Access 2000 is chosen as the database to store FamilyIS data. Features in Access 2000 is stated below:

a) Tables are the basic building blocks of databases, they are where the actual data resides.

b) Forms create framework for representing of entering data in one or mire tables. In Access, forms also have special abilities for manipulating and verifying data not available at the table level.

c) Queries search and retrieve data from one or more tables based on entered criteria.

d) Reports are a way to output data from tables or queries. Reports can summarize data.

e) Macro are a simple way to coordinate operations in Access.

3.7.5 Others Concept Used

Structure Query Language (SQL)

SQL was the basic database query language that is used for the system. It is simple to use and most importantly, it is well supported by the SQL server and other Microsoft Technologies. The simple command like select, update, insert and delete can be used to select, updated and deleted data from the database.

3.8 RUNTIME ENVIRONMENT

The Family's Information System will be a stand-alone system that needed the following runtime environment:

3.8.1 Hardware Requirement

The hardware requirements for installing FamilyIS are summarized in Table 3.2. :

Component	*Description*
Microprocessor	IBM compatible machine with Pentium 133MHz processor or higher
RAM	> 16 MB RAM
Storage	40 MB of hard disk
Input Device	Mouse and keyboard
Video Monitor	EGA, VGA or compatible display

Table 3.2 Hardware Requirements

3.8.2 Software Requirement

The software requirements for installing FamilyIS are summarized in Table 3.3:

Component	*Description*
Operating System	Windows 95 or later in standard
Programming Language	Visual Basic Enterprise Edition 6
Database	Microsoft Access 2000

Table 3.3 Software Requirements

3.9 SUMMARY

This chapter includes all the analysis done to gather information about the functional and non-functional requirements for Family Information System. In the following chapter, the database structure as well as the interface design of the system will be highlighted.

CHAPTER FOUR

SYSTEM DESIGN

CHAPTER 4 SYSTEM DESIGN

4.1 INTRODUCTION

Design is the creative process or transforming the problem into a solution; the description of a solution is also called design. The goal of system design is to translate the requirements defined during the system analysis phase into a model or representation of an entity that will be built later. During this phase, quality is fostered.

The design of FamilyIS is mainly based on the 'top down' approach. Firstly, the database is designed, followed by the program design and finally the user interface design. Figure 4.1 shows the FamilyIS's context diagram. A context diagram is a top-level diagram that contains a single process where input sources and the output destination are acknowledged. This diagram is really a bird's eye view of data movement in the system and the broadest possible conceptualisation of the system.

4.2 SYSTEM FUNCTIONALITY DESIGN

Diagram O

More detail than the context diagram permits is achievable by "exploding the diagrams". Input and outputs specify in the context diagram remain constants in this diagram. It may include more than 9 processes. The effect of drawing this diagram is that of taking a magnifying glass to view the original data flow diagram. Figure 4.2 shows the Level 0 of DFD for FamilyIS.

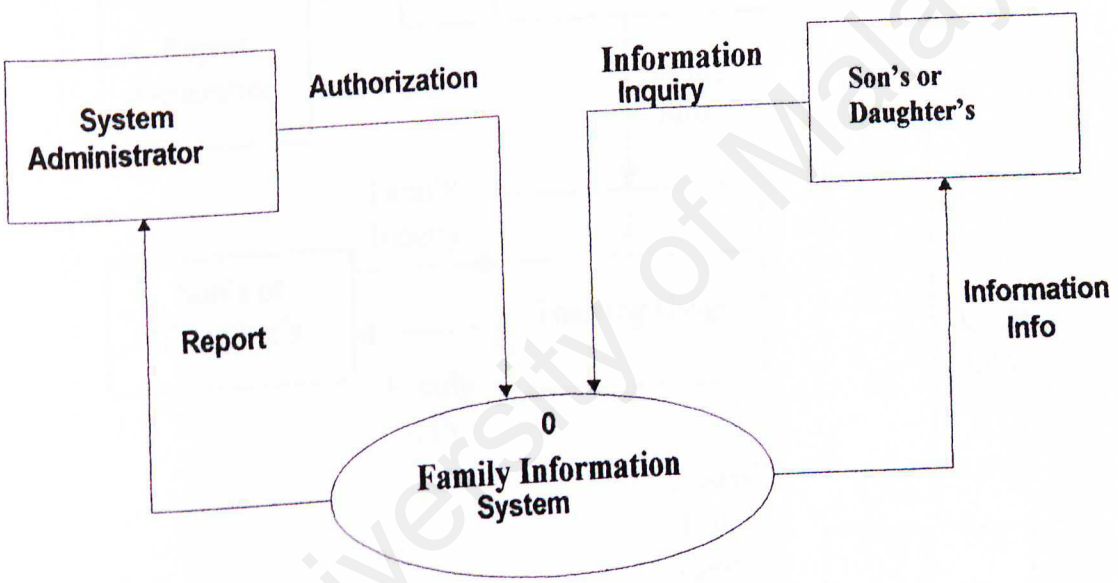


Figure 4.1 FamilyIS Context Diagram

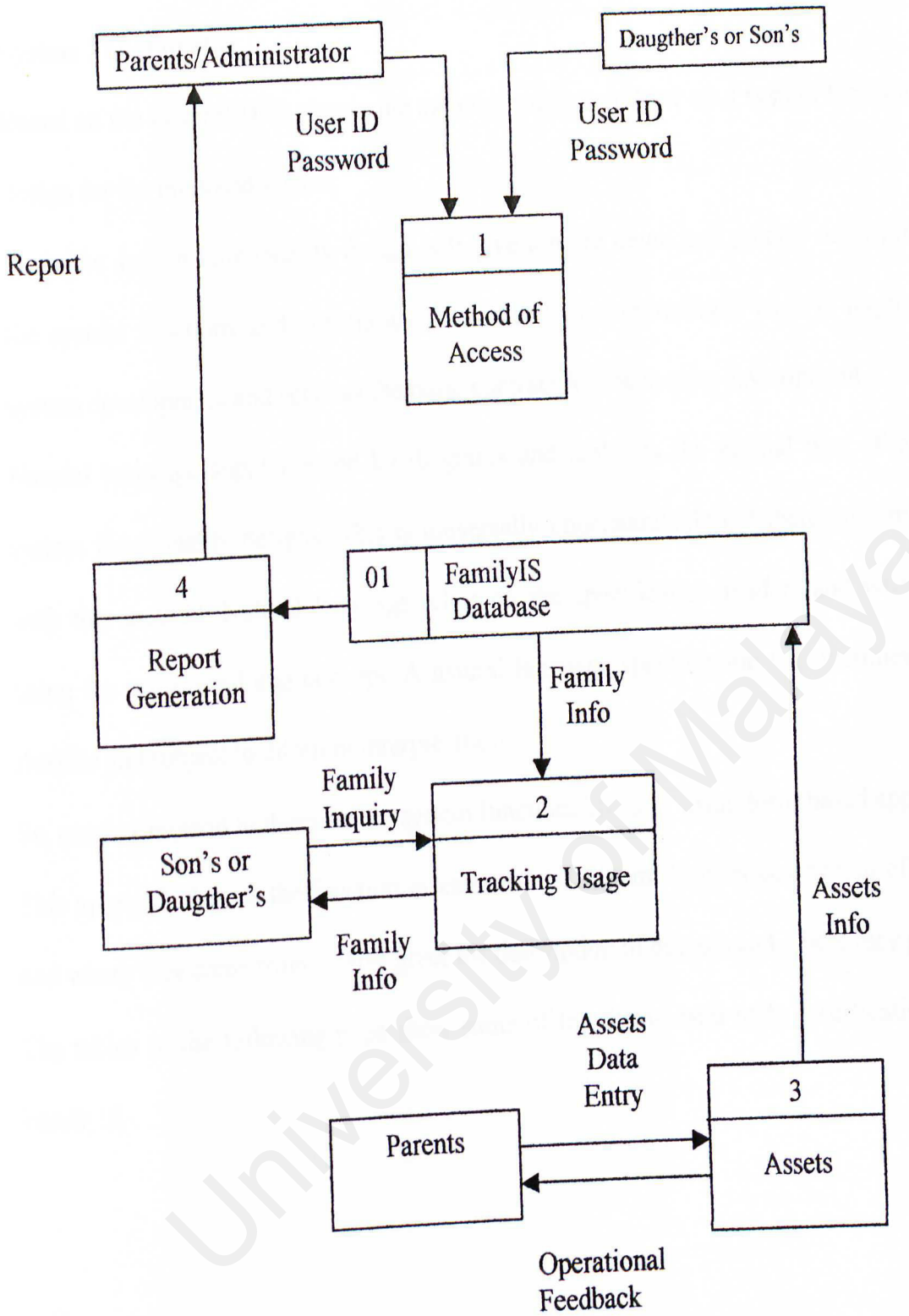


Figure 4.2 Level 0 Data Flow Diagram

System Functionality

Based on the Level 0 DFD above, the system analyst can draw up a system functionality design for the proposed system.

Here, the system functionality design will give a more detail and precise description of the system functions and constrains. It is intend to communicate what is required to system development and serve as the basic contract for the system development.

Natural language supplemented by diagrams and tables is the normal way of writing system functionality designs. This is universally understandable but there are problems with this method. Natural language relies on the specification readers and writers by using the same word and concept. A natural language specification is sometimes over-flexible and subject to different interpretation.

So, another method to describe the system functionality is by using form-based approach. This approach defines the function or entity of the system. It gives description of inputs and where they come from. It also gives the description of outputs and where they go.

The tables in the following page show some of the form-based node specifications for FamilyIS.

Function	**Method Of Access**
Description	Only valid user will be allowed to access FamilyIS.
Input	UserID and Password
Source	**TblAccess** Table
Output	Verification to access the modules in the system.
Destination	None
Requirement	User has been register as a valid user.
Pre-requisite	Splash Screen
Post-requisite	Other screens.

Table 4.1 Module Functionality for Method Of Access

Function	**Tracking Usage – By Name**
Description	To track information of families in the family.
Input	Usage_name or Item_name respectively.
Source	**User** Table, **Item** Table
Output	User_Code, User_Name, Item_Code, Item_Name
Destination	A screen to show the respective result.
Requirement	Data in the respective tables.
Pre-requisite	Tracking Usage Button
Post-requisite	Records already in the respective tables.

Table 4.2 Module Functionality for Tracking Usage-By Name

Function	**Tracking Usage – By Item**
Description	To track usage of item in the family.
Input	Item_Name respectively.
Source	**Item** Table, **Property** Table
Output	Item_Code, Item_Name, Property_Code, Property_Name, Property_Type
Destination	A screen to show the respective result.
Requirement	Data in the respective tables.
Pre-requisite	Tracking Usage Button
Post-requisite	Records already in the respective tables.

Table 4.3 Module Functionality for Tracking Usage-By Item

Function	**Assets**
Description	To keep inventory record on assets and equipments in the house
Input	Respective details.
Source	Respective details.
Output	An operation message.
Destination	**Assets** Table, **Equipment** Table
Requirement	The need to add new record.
Pre-requisite	None.
Post-requisite	None.

Table 4.4 Module Functionality for Assets

4.3 DATABASE DESIGN

A database is an integrated collection of logically related data stored in different types of records, and in a way that makes them accessible for multiple applications.

Database design involves identifying the user data requirements and determining how these data should be structured from these requirements. It transformed the unstructured information and the processing requirements of an application into representations that define the functional specifications. The database model used for this system is the relational model.

In the following section, the normalisation process in designing the database will be highlighted.

4.3.1 Normalisation

Normalisation is the technique used either to convert a large database into a relational database or create a stable, well-formatted database from scratch. Data normalisation provides rules on how to break tables or field into several tables that has references to each other. Data normalisation seeks to minimize duplication of data within the database by logically dividing a large table into several smaller tables. This process of normalisation makes information is more differentiated and usable. The main purposes of normalisation are to reduce data redundancy and to eliminate data inconsistency. A properly normalised database will not only save storage but also minimize the need to modify data relations if the system is later expended or upgraded.

The steps below describe the basic normalisation process.

Step 1: First Normal Form (INF)

This step involves removing the entire repeating group and identifying the primary key(s).

Step 2: Second Normal Form (2NF)

The second step involves removing all partial functional dependencies by splitting the original relation into more relations. The relation is said to be in 2NF is the relation that is in 1NF and every non-key attribute is fully functional dependent on the primary key.

Step 3: Third Normal Form (3NF)

The third involves removing any transitive dependencies where non-key attributes are dependent on another non-key attributes. The relation is said to be in 3NF if the relation is in 2NF and there is no transitive dependency.

4.3.2 Data Dictionary

As most of us know, the volume of data in most of applications is substantial (more than a single analysts can easily keep track of). When teams of analysts work on a system, the task of coordinating data definition becomes more complex. Therefore, a data dictionary has to be developed in order to let system analysts and programmers to keep track of data definition used in the system. Individuals depend on the definitions other established and the assumption they made about data specification. A data dictionary is a repository of elements in a system. As the name suggest, these elements centre on data and the way they are structured to meet user requirements and organisational needs. In a data dictionary, a list of all the elements composing the data following a system can be found. The major elements are data flows, data stores and process. The data dictionary stores details and description of these elements.

There are 5 reasons why data dictionary are important.

i) To manage the detail in a large system.

ii) To communicate a common meaning for all system elements.

iii) To document the features of the system.

iv) To facilitate analysis of the details, in order to evaluate characteristic and determine where system changes should be made.

v) To locate errors and omission in the system.

The following are the specification of each table in FAMILYIS database.

Table Name : Family		
Description : Contains information on the family in house.		
Column Name	**Data Type**	**Description**
*Family_Code	Text	Family code.
Family_Name	Text	Actual name of the Family.
Family_Desc	Text	Special description of the Family.
Schedule_Flag	Yes/No	To see whether the Family already been schedule for any activities

Table 4.5 Family Table

Note: * Primary Key

Table Name : Assets		
Description : Contains information on asset or special assets in the house.		
Column Name	**Data Type**	**Description**
*Asset_Code	Text	Asset code
Asset_Name	Text	Actual name of he asset
Asset_Capacity	Number	Capacity of the asset.
Location_Code	Text	Location where the asset are located.
Asset_Desc	Text	Special description of the asset.
Usage_Code	Text	Usage code.

Table 4.6 Asset Table

Note: * Primary Key

Table Name : Vendor		
Description : Contains information on the vendors who sell assets to the family.		
Column Name	**Data Type**	**Description**
*Ven_Code	Text	Vendor's code
Ven_Name	Text	Vendor's name
Con_First_Name	Text	Vendor's first name.
Con_Last_Name	Text	Vendor's last name.
Con_Title	Text	Position of the contact person, for example Manager, Sales Person, etc.
Ven_Addr1	Text	Vendor's address 1
Ven_Addr2	Text	Vendor's address 2
Ven_Addr3	Text	Vendor's address 3
Ven_Postcode	Number	Postcode
Ven_Telno	Number	Telephone number.
Ven_Faxno	Number	Fax number.
Ven_Email	Text	E-mail address.
Ven_Desc	Text	Other's description about the vendor

Table 4.7 Vendor Table

Note: * Primary Key

Table Name : Asset_Type		
Description : Contains information on types of asset in house.		
Column Name	**Data Type**	**Description**
*Type_Code	Text	Type code that will be use to generate asset no.
Type_Name	Text	Asset type name.

Table 4.8 Asset_Type Table

Note: * Primary Key

Table Name :	Usage	
Description :	Contains information on the usage of every assets in a Family	
Column Name	**Data Type**	**Description**
*Usage_Code	Text	Usage code.C, S
Usage_Name	Text	Name of the usage, C-Common, S-Special

Table 4.9 Usage Table

Note: *Primary Key

Table Name :	Report	
Description :	Contains information on assets and tracking information.	
Column Name	**Data Type**	**Description**
Report_ID	Auto Numbew	An ID to uniquely identify the report done.
Report_Date	Date/Time	Date the asset was purchased
UserID	Text	To keep track of who report the asset
Report_Desc	Text	Description of the particular asset
Location_Name	Text	Place of the asset.
Fixed	Yes/No	To see whether this record has been fixed or not.
System_time	Date/Time	To keep the current time the report being made.
Report_Code	Text	Code of the asset.

Table 4.10 Report Table

Table Name : cpr		
Description : Change Password Record		
Column Name	**Data Type**	**Description**
* cpruid	Text	User identity key
cprupw	Text	User password

Table 4.11 cpr Table

Note: * Primary Key

Table Name : Upf		
Description : User Personal File - store user details		
Column Name	**Data Type**	**Description**
* upfuid	Text	User identity key
Upffnm	Text	User full name
Upfuic	Text	User identity card number
Sexcod	Text	User gender
Upfhpn	Text	User phone number
Upfadd	Text	User address
Utfutn	Text	User type name
Upftdd	Date/Time	The year of birth for the user
Upfdel	Yes/No	Deleted flag
Poscod	Text	User post in family

Table 4.12 upf Table

Note: *Primary Key

Table Name : Utf		
Description : User Type Facility Table – store the user type of facility.		
Column Name	**Data Type**	**Description**
*utfutc	Text	User type code, A, T, S
Utfutn	Text	User type name P- Parents/Administrator, Son's or daughter

Table 4.13 utf Table

Note: * Primary Key

interface technology. Therefore, the system being designed should take into consideration the skill level and behaviour of users.

There are two categories of the user interface design guidelines that can be followed. They are general interaction and information display.

4.4.1 General Interaction

i) Be consistent

Used a consistent format for menu selection, command input, data display and the myriad other functions that occur in an user interface.

ii) Offer meaningful feedback

Provide the user with visual feedback to ensure that two-way communication (between the user and interface) is established. For instance, the mouse pointer will chance its shape to an hourglass shape to visually inform the user that the system is now busy in process.

iii) Ask for verification of any non-trivial destructive action

Always prompt user for actions such as deletion of a record before the action is carried out. Before an item or user is deleted from the database, a message box will appear in front of the user for confirmation.

iv) Forgive mistake

The system should protect itself from user's errors that might cause it to fail.

4.4.2 Information Display

i) Display only information that is relevant to the current context

The user should not have to wade through extraneous data, menu and graphics to obtain information to a specific system function. For example, only needed user information is displayed in the user list screen.

ii) Use consistent labels, standard abbreviation and predictable colours

The meaning of a display should be obvious without reference to some outside source of information. In FAMILYIS, a standard and consistent command button for closing a screen is used. This command button has the same caption and icon on it in every screen that it appears.

iii) Inactivated commands that are inappropriate in the context of current

This is to prevent the user from attempting some action that could result in error. If the user does not have access rights to some commands or buttons, the system will hide it from their view.

4.4.3 Menu Design

A menu interface provides the user with an on-screen list of available selection. In responding to the menu, user is limited to the options displayed. The user need not know the system but does need to know what task should be accomplished.

The following figure (Figure 4.2) and table (Table 4.10) gives the description of the menu bar in FAMILYIS.

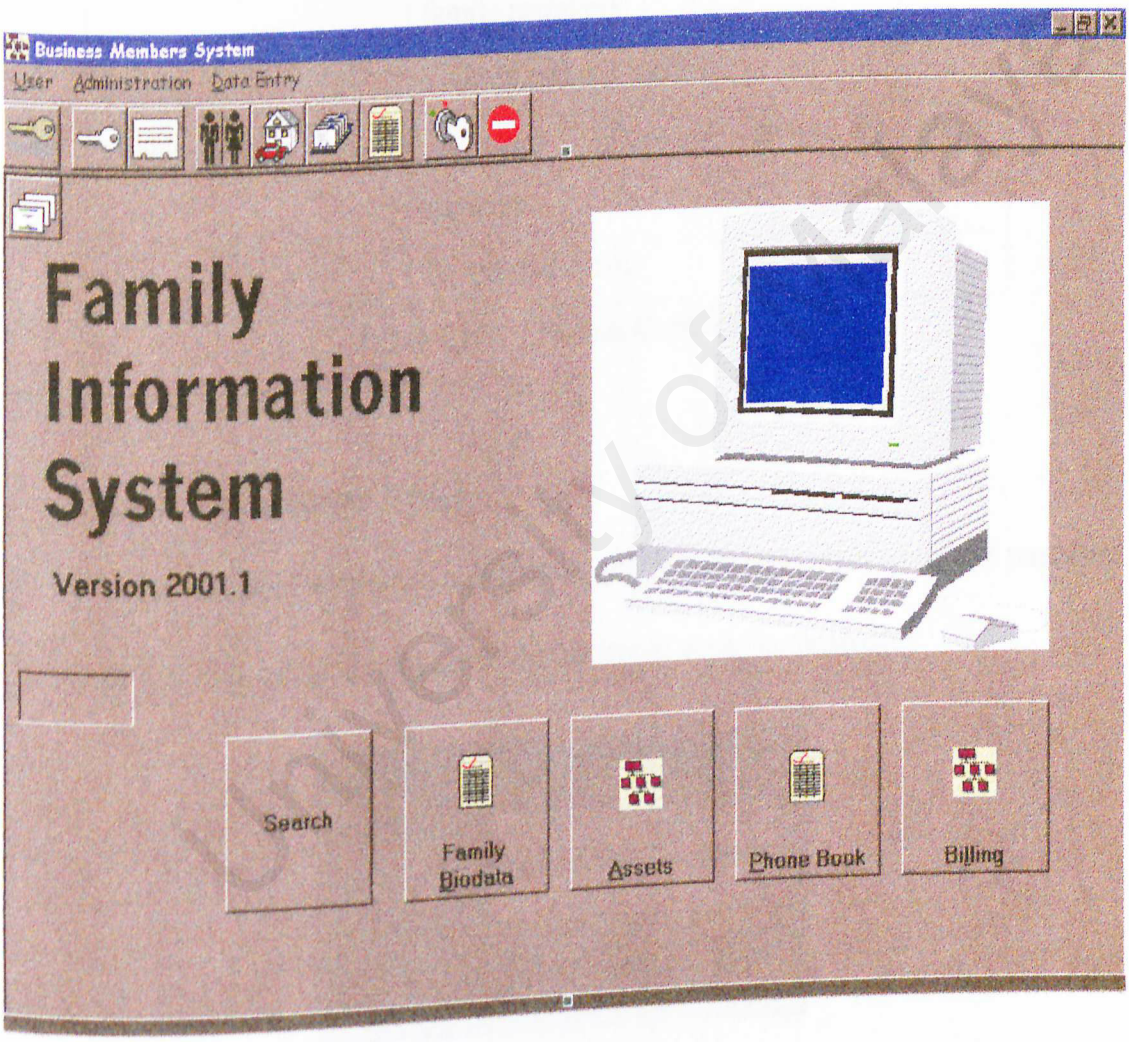


Figure 4.3 Main menu interface

Button	Description
Search	To track usage of families either by assets or by phone.
Report	To generate report for house administrator.
Assets	To keep assets records.
Administration	To add in new user by the system administrator.
Family Biodata	To keep record of the personal details of family members
Billing	To keep billings record.
Phone Book	To show information about the phone numbers.
Exit	To exit the system.

Table 4.14 Button Description

4.4.4 Login Screen Design

Login screen is where the user need to key in their user name and their password so that they can gain accesses to the system as well the database.

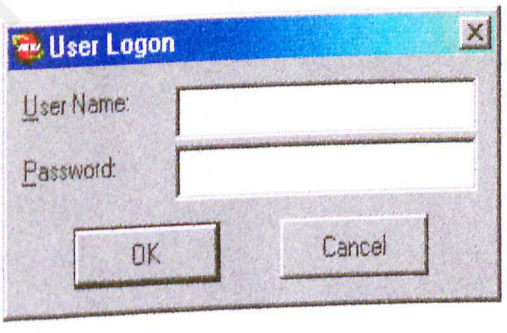


Figure 4.4 Login screen

4.4.5 Miscellaneous Screen Designs

Below are others screen designs for the FAMILYIS. They are generated based on the guidelines and description above.

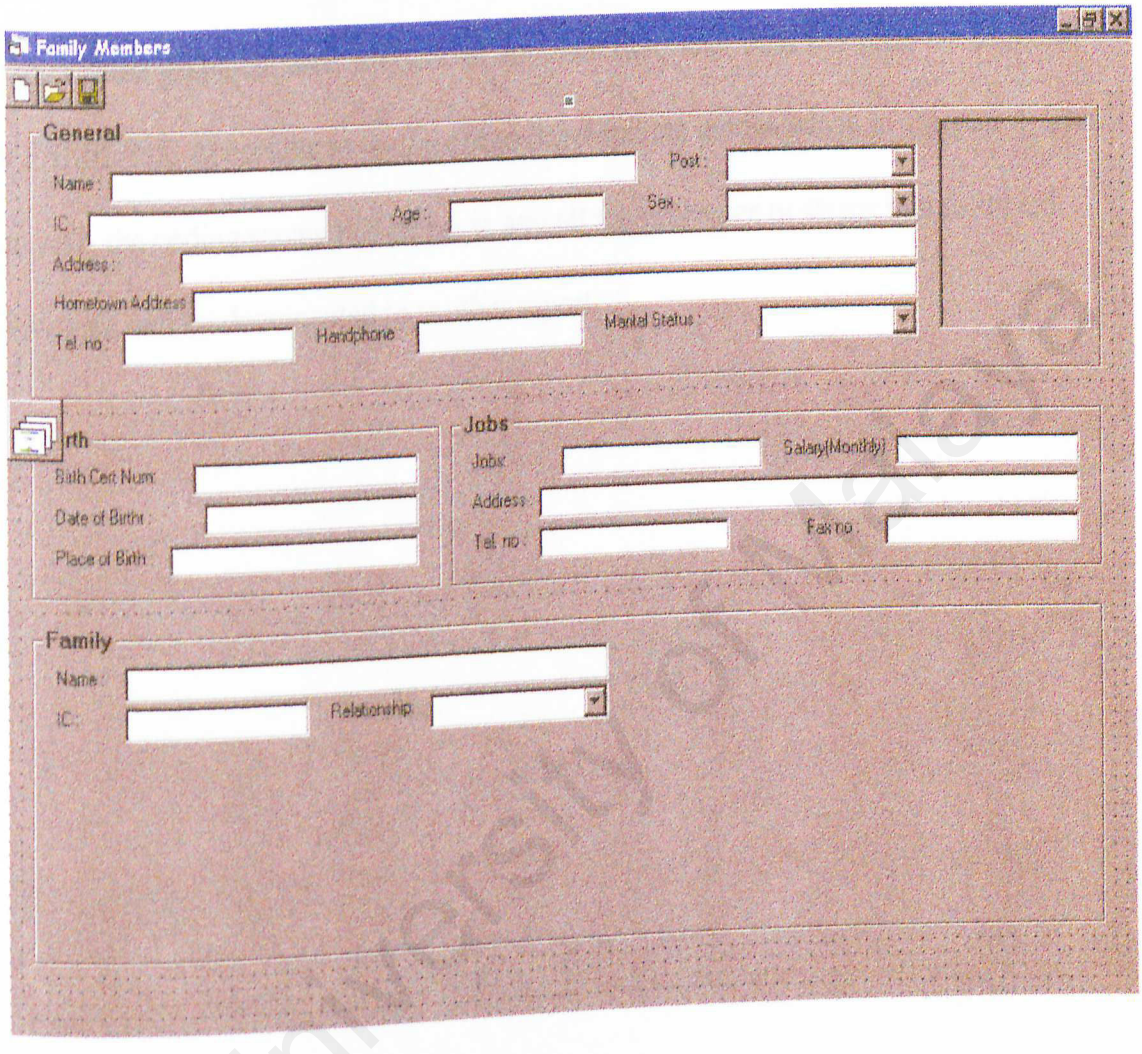


Figure 4.5 Family Biodata

4.5 SUMMARY

Functional specifications of the main modules or function for FAMILYIS are being defined here in this chapter. It also includes the design of database. Also included in this chapter is the graphical user interface of the main functions. And this ends the proposal for FAMILYIS. According to the proposed development methodology in Chapter 2, later this year, the coding stages will be begin and all the processes or stages that followed afterwards will be documented in another report.

CHAPTER FIVE

SYSTEM IMPLEMENTATION

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation is the physical realization of the database and application design. On completion of the design stages (which may or not may not have involved prototyping), here comes the stage where the database and the application have to be implemented. The transaction process continues when a compiler accepts source code as input and produces machine-dependent object-code as output. Compiler output is further translated into machine code – the actual instruction that drives micro-coded logic in the central processing unit (CPU).

Coding and debugging are the major works involved in the implementation phrase. So, coding mythology, documentation and testing will later be covered in this chapter or in later chapters.

5.2 CODING

Program written must be able to implement the design. This task can be daunting, for several reason. First, the designers may not have addressed all of the idiosyncrasies of the platform and programming environment; structure and relationship that are easy to describe with charts and tables are not always straightforward to write as code. Second, a programmer must write out code in a way that is understandable. Third, the programmer must take advantage of the characteristic of the design's organization, the data structure, and the programming language's constructs while still creating code that is easily reusable.

The coding methodology used in the development of this system is the top-down and bottom-up approach. By combining both approaches at different stages of coding, testing could be done on those completed modules while others are still being coded. Comments are added in the code to provide a better understanding to the code.

5.2.1 Top-down approach

This approach allows the higher-level modules to be coded first before the lower level modules. The codes in the lower modules contain only an entry and an exit. A module with such characteristic is called a shell. The higher-level modules will reference the lower ones if they are coded and available. Reference to a shell will result in an empty action.

This approach will ensure that the most important modules will be developed and tested first. It also gives a preliminary version of the system sooner.

Below is a diagram that describes the top-down approach for FamilyIS.

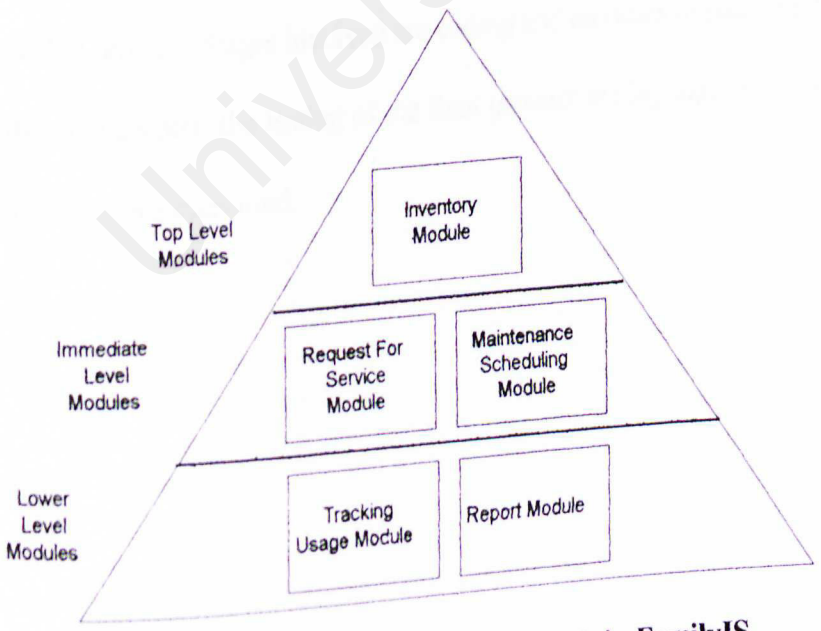


Figure 5.1 Top-down Approach in FamilyIS

5.2.2 Bottom-up approach

As oppose to the top-down approach, the bottom-up approach begins with the coding of the lower level modules first before the higher level modules. However, the higher modules are just skeletons that call the lower modules. This approaches is used if the critically of lower level modules is high and need to be completed first.

5.3 MODULE IMPLEMENTATION

The Family Information System is divided into 7 main modules, which are the Biodata Module, Assets Maintenance Scheduling Module, Billing Module, Accounts Module, Report Module, and Administrator Module. Each module is developed using Visual Basic 6.0. (Chapter 3 provide more explanation of Visual Basic 6.0)

5.4 SUMMARY

Under this stage, the design model of the Family Information Systems are being into a workable product. Stages involved are coding and modules implementation.

In the next chapter, the testing of the final product are highlighted. It will show types of testing approaches used.

CHAPTER SIX

SYSTEM TESTING

CHAPTER 6 SYSTEM TESTING

6.1 INTODUCTION

Testing is the process of executing the application programs with the intent of finding errors. Before going live, the newly developed database application system should be thoroughly tested. This is achieved using carefully planned test strategies and realistic data so that the entire testing process is methodically and rigorously carried out. In fact, testing cannot show the absences of faults, it can show only that software fault. To quote Deutsch [DEU79],

The development of software system involved a series of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives... may be erroneous or imperfectly specified, as well as [in] later design and development stages... Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation.

In this chapter, software testing fundamentals, testing strategies and software debugging methods will be presented.

Following are some of the objectives of software testing:

i) Testing is a process if executing a program with the intent of finding an error.

ii) A good test case is noted that has a high probability of finding an as-yet-undiscovered error.

iii) A successful test is one that uncovers as-yet undiscovered error.

6.2 Testing Technique

To test a component, a range of inputs and conditions are chosen. The component of the software will be allowed to manipulate the data, and the output will be observed. A particular input is chosen will demonstrate the behavior of the code behind the entire GUI. A test point or a test case is a particular choice of input data to be used in testing program. However, the data are entered with the express intent of determining whether the system will process them correctly.

Different test cases are needed on different type of testing strategies. As there is no available end-user, artificial test data that resembles the actual facilities in school are created. There are three categories of test cases that are created for testing purposes namely erroneous test, normal test data, and extreme test data. These categories are further explained in the following section.

6.2.1 Erroneous test

Using test data that are erroneous is a good way to determine how the system handles such errors and how it behaves under such situation. For example, an invalid user will not be allowed to enter FamilyIS until he or she gives a right combination of username and password. For this case, the system will prompt the user "invalid username or password". Therefore, an invalid username or password is use as erroneous test data.

6.2.2 Normal test data

The normal test case is use to check whether the system will work well under normal condition. One example of the normal test data is shown below.

Assume there are only three data in January. There are another four data in February. So, if a user want to print a report on the total amount, the system will pick up seven records in both months. The expected results can be compared both on the screen as well as on the hard copy.

This type of test data serves as a preliminary test of the system.

6.2.3 Extreme test data

The extreme test data is use for exhaustive testing. The test data can be very huge or it can be at or beyond the boundary value. Lets take another example.

The system does not allow the user to choose the date of a malfunction of an asset bigger than the current system date. This is because, how can a user know that the asset will be malfunction on a date in the future?

6.3 TESTING STRATEGY

A strategy use to test this system (FamilyIS) is actually a series of four steps that are implemented sequentially. The steps are shown in Figure 6.1. Initially, tests focus on each component individually, ensuring that it functions properly as a unit. Hence, the name *unit testing*. Unit testing makes a heavy use of white – box testing technique, exercise specific paths in a module's control structure to ensure complete average and maximum error detection. Next, components must be assembled or integrated to form the complete software packages. Integration testing addresses the issues associated with the dual problems of verification and program construction. Black-box test case

design techniques are the most prevalent during integration, although limited amount of white-box testing maybe use to ensure coverage of major control path. After the software has been integrated (constructed), sets of high-order tests are conducted. Validation criteria (established during requirement analysis) must be tested. Validation testing provides final assurance that software meets the functional, behavioral, and performance requirement. Black-box testing techniques are used exclusively during validation.

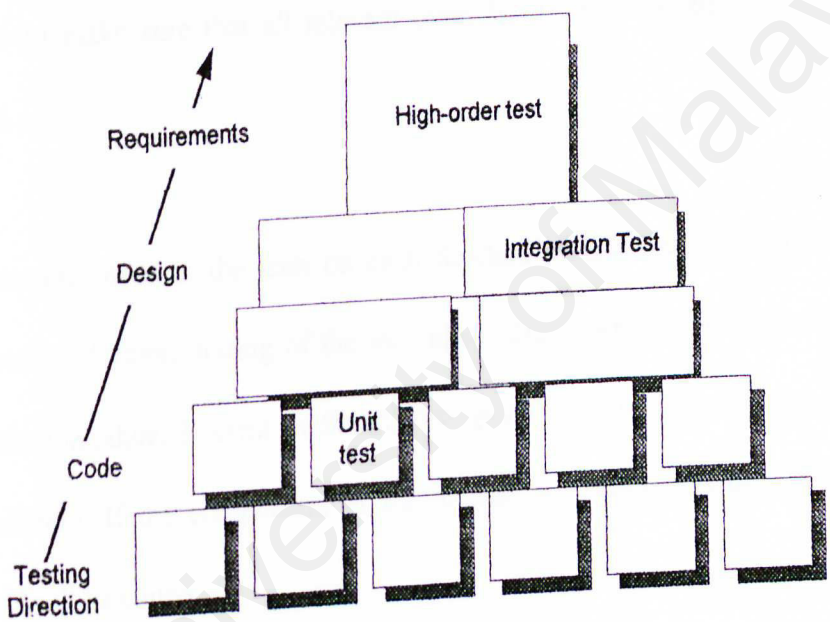


Figure 6.1 Software Testing Steps

The last high-order testing step falls outside the boundary of software engineering and into the broader context of computer system engineering. Software, once validated, must be combined with other system elements (e.g., hardware, people, databases). System testing is an example of high-order test, verifies that all system elements mesh properly and that overall system function/perfomance is achieved.

6.3.1 Unit testing

Historically, quality software is relied on testing each function or module. Unit testing is sometimes referred as Module Testing or Component Testing, which is extremely time-consuming. For FamilyIS, unit testing was done during the coding phrase. The first step is to examine the program code by reading through it, trying to faulty algorithms or syntax faults.

The process is followed by comparing the code with specifications and with the design to make sure that all relevant cases have been converted into the desired output.

Unit testing involves the tests on each function module independently. In Family Information System, testing of the individual class module is merely compiling the individual module. If error is found, debugging of the codes will be carried out immediately. If the compilation of the module is completed successfully, another module will be coded.

6.3.2 Integration testing

Testing a specific feature together with other newly developed feature is known as integration testing. In other words, when the individual components are working correctly and meet the objectives, these components are combined into a working system. Testing the interface of 2 components explores how components interact with each other.

Incremental integration approach was applied during the developments of the system. The system was constructed and tested in small arguments, where errors were easier to isolate and correct. Error will be corrected before processing to the next integration.

If all individual modules passed the Module Testing successfully, there will be no bugs in the Module Integration Testing. The motive behind this testing is to make certain that all modules can be executed as a complete module. As mentioned earlier, an individual module calls other module to perform certain tasks. Parameters will be passes among these modules and if not tested, then parameter may be passed incorrectly.

6.3.3 System testing

System testing is a series of different tests conducted to verify that all system elements such as hardware, software and information, have been properly integrated and performed allocated function. Below are some of the testing done:

i) Performance Testing

System performance is measured against the performance objectives set by the user as expressed in the non-functional requirement. Performance testing examines the speed of response to user command, accuracy of the result, and accessibility of the data are checked against user's performance prescription.

ii) Platforms testing

Platform testing involves the testing of the Family Information System on the different platforms. Different platforms mean different operating system (Windows

CHAPTER SEVEN

SYSTEM EVALUATION AND CONCLUSION

CHAPTER 7 SYSTEM EVALUATION AND CONLUSION

7.1 INTRODUCTION

After all the handwork of designing and developing as well as implementing FamilyIS, the end product of the project is brought up for evaluation. They are many evaluation techniques that use to evaluate the final system. The following section will explain in detail about the system strength and its limitation.

7.2 PROBLEMS FACED AND SOLUTIONS

The following are the major problems encountered from the beginning of the project through the end of the system development process.

i) Difficulties in choosing a programming language and tools

There are some well-known programming tools available in the market that can be use to develop a similar database management system. (Please refer to Chapter 2). Choosing a suitable tool was a critical process as all tools have tier strength and weakness. In addition, the availability of the required tool for development was also a major concern.

Solution:

There is no latest version of Power Builder 7.0 available in FSKTM. So, I have to choose Microsoft Visual Basic 6.0.

ii) Difficulties in gathering information

Not much information could be gathered from the questionnaires that were distributed to the family. This is due to from 5 questionnaires sent to family, only 1 responded.

Solution:

Obtain more information and ideas through brainstorming with my supervisor as well as other lecturers. Ideas are also gathered from other course mates. Their ideas are also useful because we all are members in our family.

iii) Not enough end user evaluation

Since there is no really many end user to test FamilyIS except evaluation from my coursemates, I do not really know how end-user will react to FamilyIS.

Solution:

Tests were done on other group like students from other faculty and working group.

iv) Slow processing time

As very 4GL programmer knows, Visual Basic 6.0 is a graphical-oriented programming language. As such, application created using this language is also graphical-oriented and thus more memory is required to compile and execute the application.

Solution:

Although it is documented that the minimum memory requirement in developing FamilyIS is 32 MB, but personally a memory of 128 MB is much preferred.

v) Transferring SQL Statement from VB to Crystal Report

Syntax the SQL Statements in VB6 and the syntax in Crystal Report differ from each other. So, I cannot use the knowledge of my SQL statement that I learn from VB to Crystal report. This make quite hard for me to pass the selection criteria from VB6 to Crystal Report.

Solution:

Referring to online tutorial about Crystal Report and read about Crystal Report from e-books and others web-publish tutorials.

7.3 SYSTEM STRENGTHS

i) Security features

There are 2 types of users in FamilyIS: the Administrator (Parents), and the sons or daughters of the family. Each user type is allocated certain access rights to the functions in FamilyIS. For example, only the Administrator is allowed to create others user. Therefore, functions that are restricted to the user will be disabled. By incorporating these security features in FamilyIS, the possibility of an unauthorized access will be greatly reduced.

ii) Friendly user interface

FamilyIS is specially designed on the principle for ease to use. As such, GUI features have been integrated into the system. The inclusion of GUI has contributed vastly to aid users. Users can easily capture the overview of the system, without even referring to any help. The design of the GUI has been designed as simple as it is just to make

sure that the user that using the FamilyIS(mostly no computer knowledge) can understand easily and not to get confused.

iii) Request For Service

FamilyIS provides function to the user to report any malfunction or broken assets in the family so that action to repair the particular assets can be done as soon as possible.

iv) Maintenance Scheduling

FamilyIS provides function to the user to schedule maintenance of assets and accounts. A message box will prompt the user to inform him that the time to do maintenance for a particular event has arrived.

7.4 SYSTEM CONSTRAINTS AND LIMITATIONS

i) Maintenance scheduling prompting

The function that inform the user that the time to do maintenance for a particular event is done by comparing the current system date with the scheduled maintenance date in the database. However, the user can change the date and time of the system through Control Panel. FamilyIS unable to sense changes of system date and time.

ii) Single host

FamilyIS is a stand-alone system and therefore does not support the multi-user environment. In order to use the system, user has to install the FamilyIS into their PC

7.5 FUTURE ENHANCEMENTS

i) Dual lingual support

Due to the time constrain in developing FamilyIS, the language use for the entire interface is in English and Bahasa Malaysia. If the system could provide other language support like Chinese etc, it would be more fascinating because some of the administrative works in family are done in other language.

ii) Interactive help (in demo or video from)

If a system could provide a demo package or a short tutorial session, it will definitely help the users learn up the system in a shorter period.

iii) Networking

To really benefit all the users, FamilyIS shall be modified to enable network accessing. This can be achieved by using the LAN (Local Area Network) environment. In a multi-user environment, FamilyIS only needed to be install in the server machine and other client machines can gain access to FamilyIS, simultaneously.

7.6 KNOWLEDGE AND EXPERIENCE GAINED

i) Microsoft Visual Basic 6.0 (Database Programming and Event-Driven Programming)

Developing this system has given me the opportunity to learn VB6. Reading about VB6 is not enough to understand its characteristic. By developing this system, I learn more about VB6 characteristics and these characteristics cannot be gain from reading books.

ii) Microsoft Access 1997

The database in FamilyIS is developed using Access 1997. I have the opportunity to learn more about Access and how to connect to VB6. Only through developing this system that I learn that VB6 cannot be connected to Microsoft Access 2000. So, I have to convert the database to Access 97 format.

iii) Self Expression

Developing FamilyIS has really given me a great change to express myself in designing and coding of the system. Finally, before graduating, I have the change to build application software by myself. Doing this thesis has greatly improved my self-esteem and self-confidence.

7.7 SUMMARY

Evaluation of a system is indeed needed to ensure its objectives and intended functions have been achieved. This chapter covers all the aspect of evaluating application software. At the end of evaluation, comes the conclusion of this thesis project.

7.8 CONCLUSION

The Family Information System is a start to a computerized family environment in Malaysia. FamilyIS will be an example of what the function of information management in family will be like.

To be given a change to develop this system has been both enjoyable and frustrating to me. It is enjoyable because finally I can really manipulate and use all the

knowledge gathered during the three-year course in FSKTM, particularly on the use of software engineering, system analysis and design as well as database management.

It is frustrating because I have to learn the tools that I have chosen (VB) by myself. Although there is a lot of online VB tutorials, however choosing a website that is really suitable is a task for me. So, I would like to take this opportunity to suggest to the faculty to offer a course to teach the students one of the 4th generation languages in the market for example Visual Basic, Power Builder or others. This course will definitely help the student later in their studies or in their work.

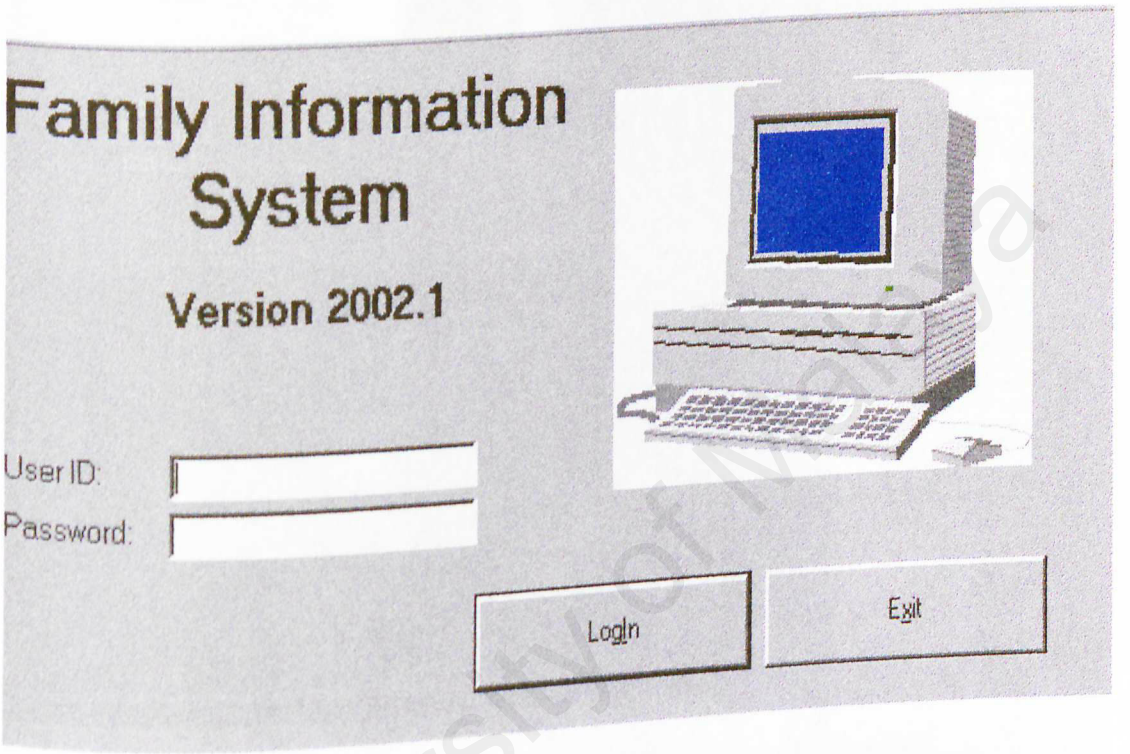
Last but not least, the FamilyIS can be said to have achieved its objectives as well as it requirement as planned earlier. It will serve well as a computerize facilities management in typical family in Malaysia.

APPENDIX

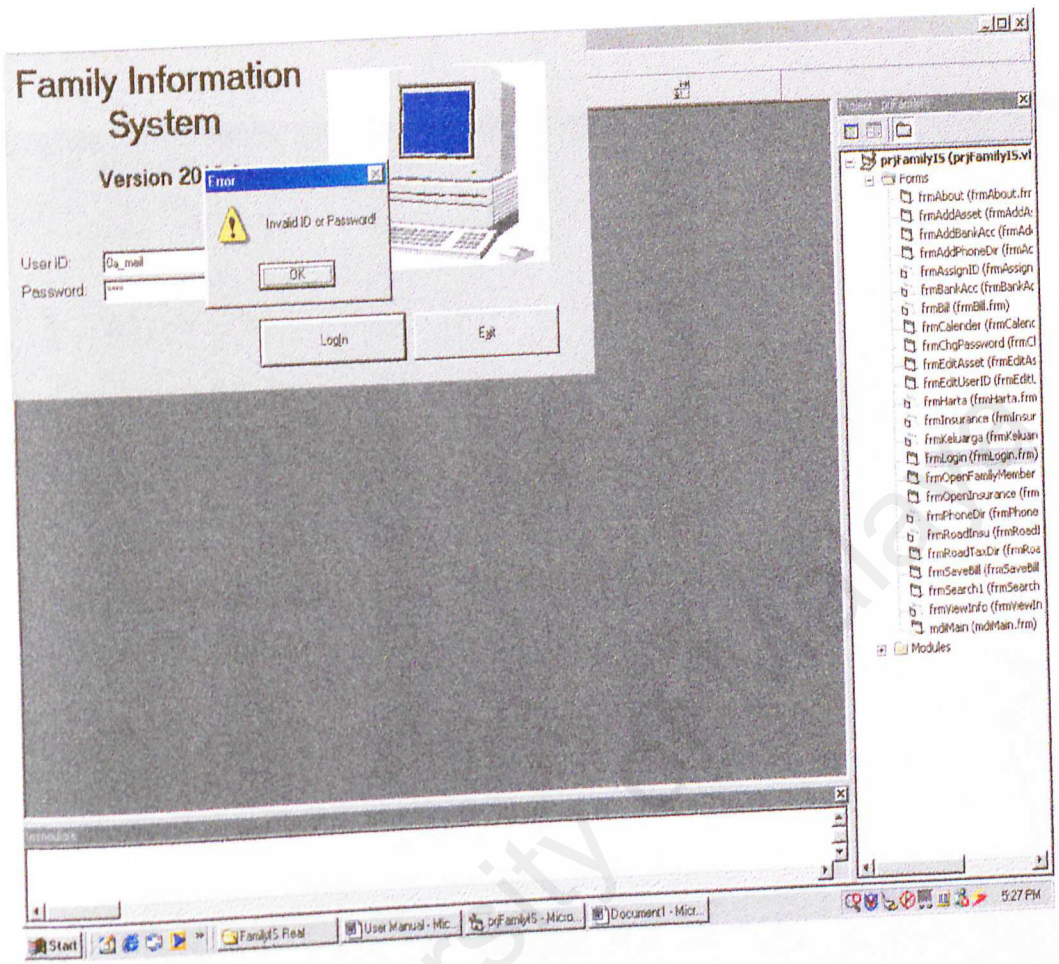
USER MANUAL

USER MANUAL FOR THE FAMILY INFORMATION SYSTEM

The Login Screen

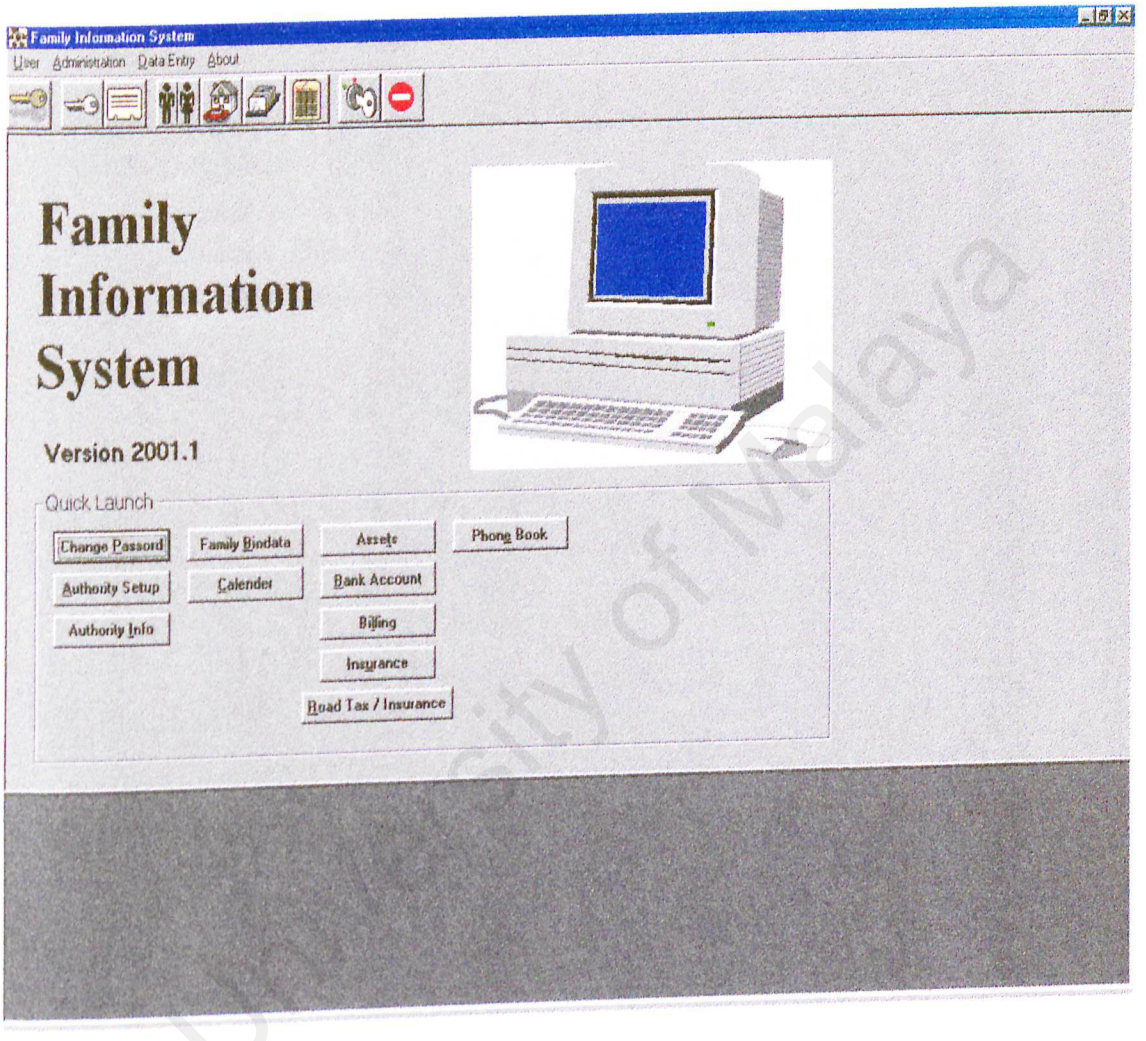


This screen will be the first screen to appear in the system to validate the user.



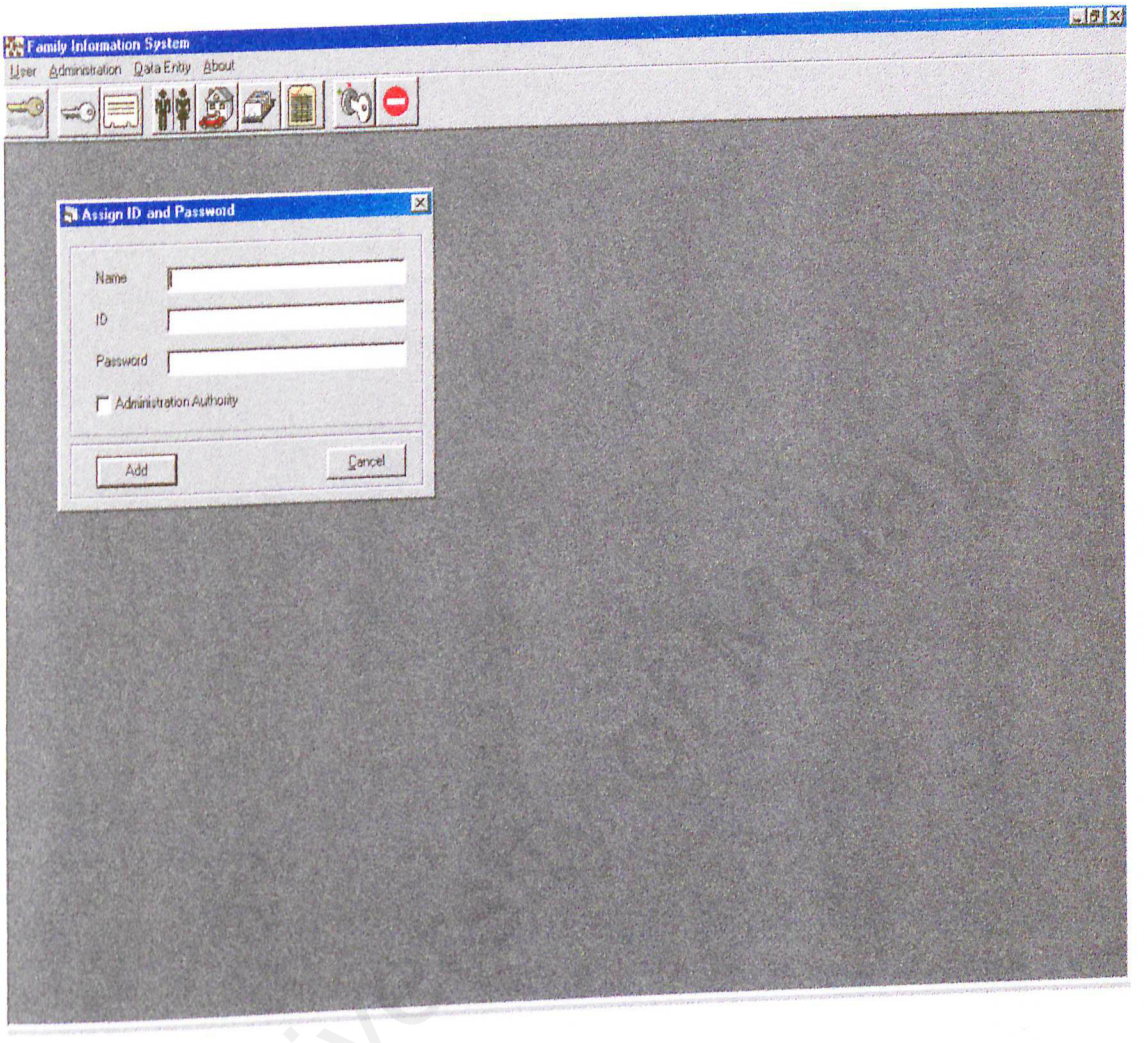
Message for the invalid username or password.

Main Screen



Using this main screen, user can choose whatever function that they want to execute. For example, if a user want to key-in the latest bill, he or she can do so by clicking on the Billing command button.

Add User

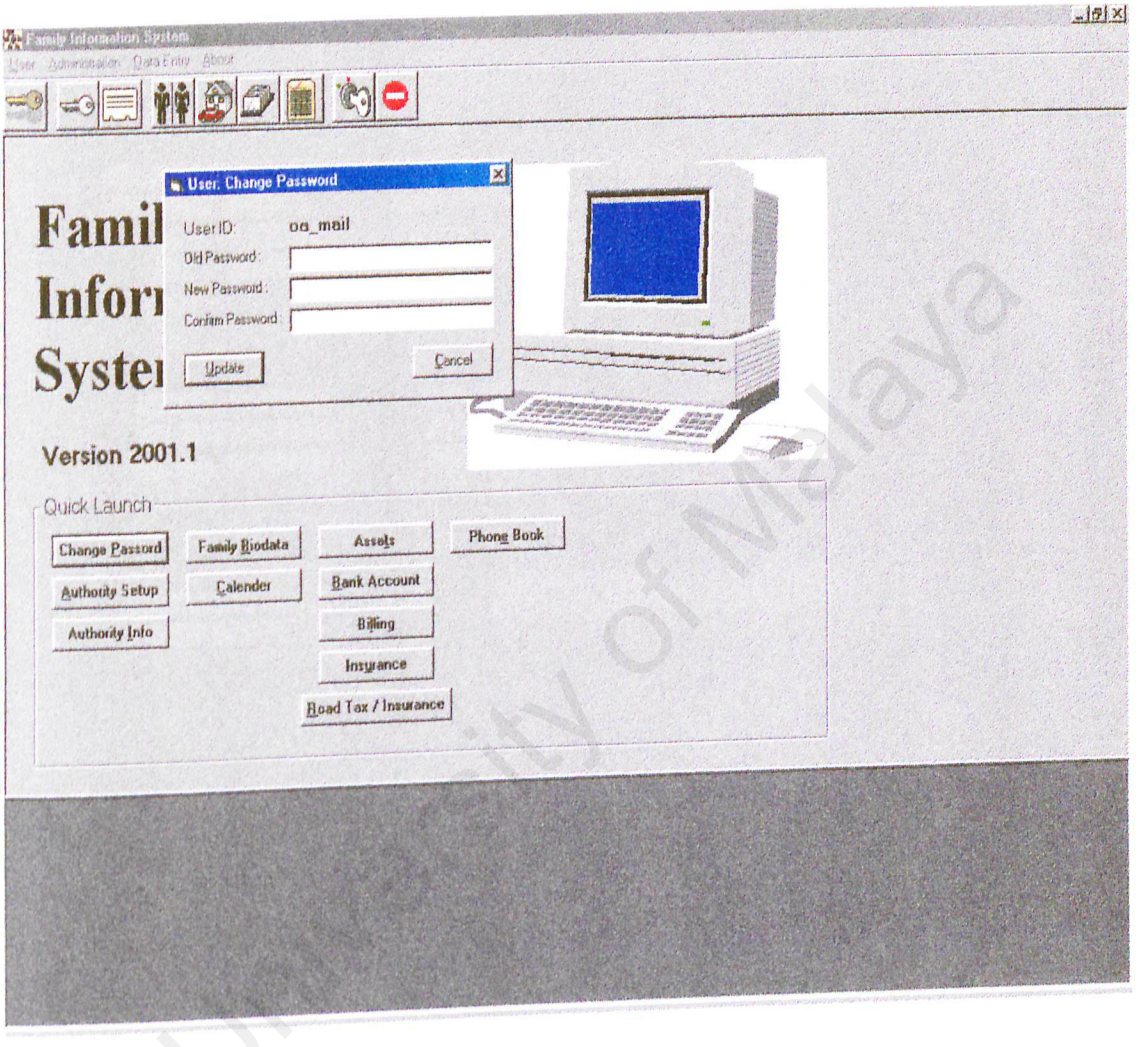


This screen enable the family member to add additional members into the system.

This screen also enable the user to specify whether an added user to have an

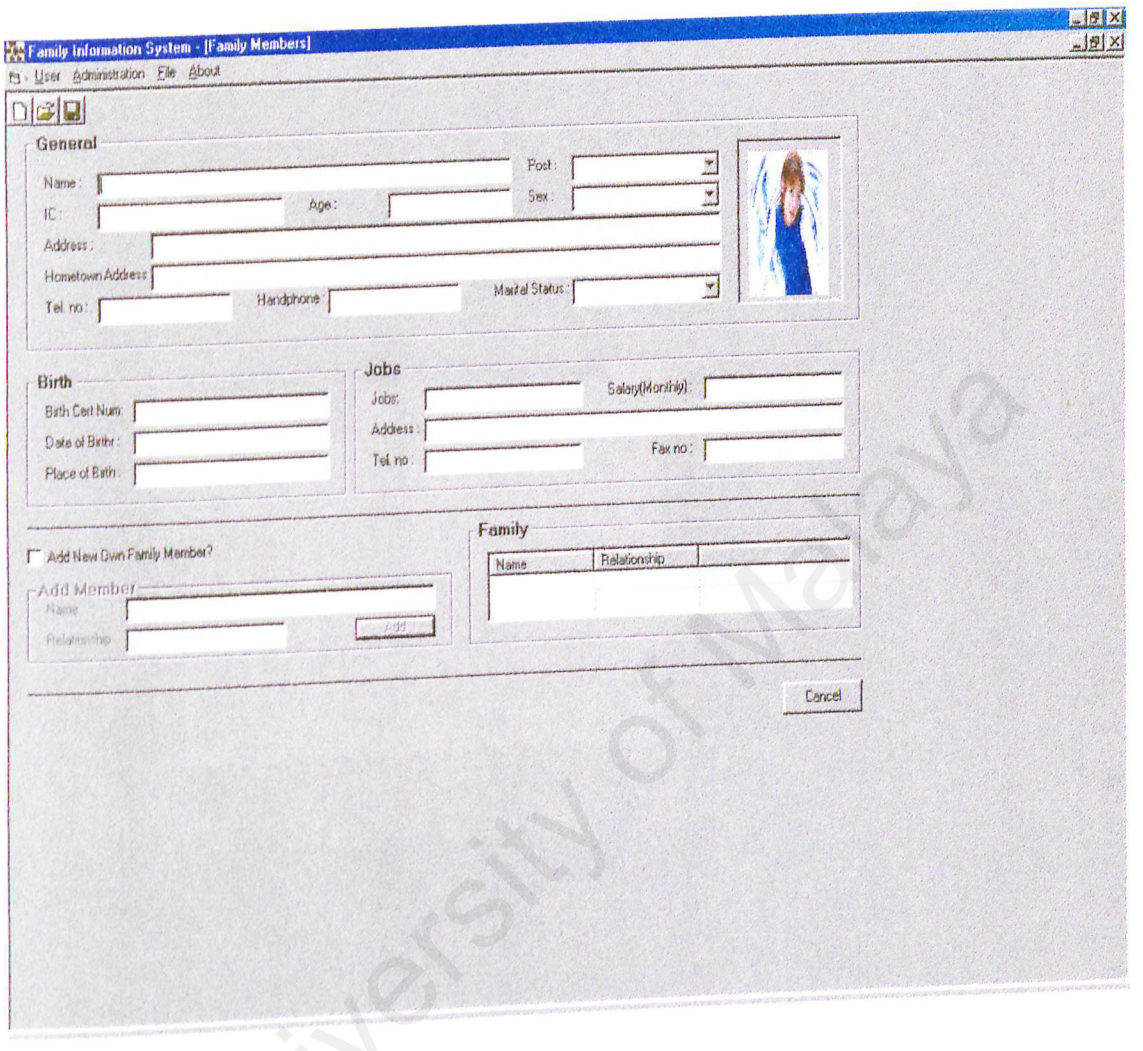
administrative authority or not.

Change Password



This screen enable a user to change their respective password.

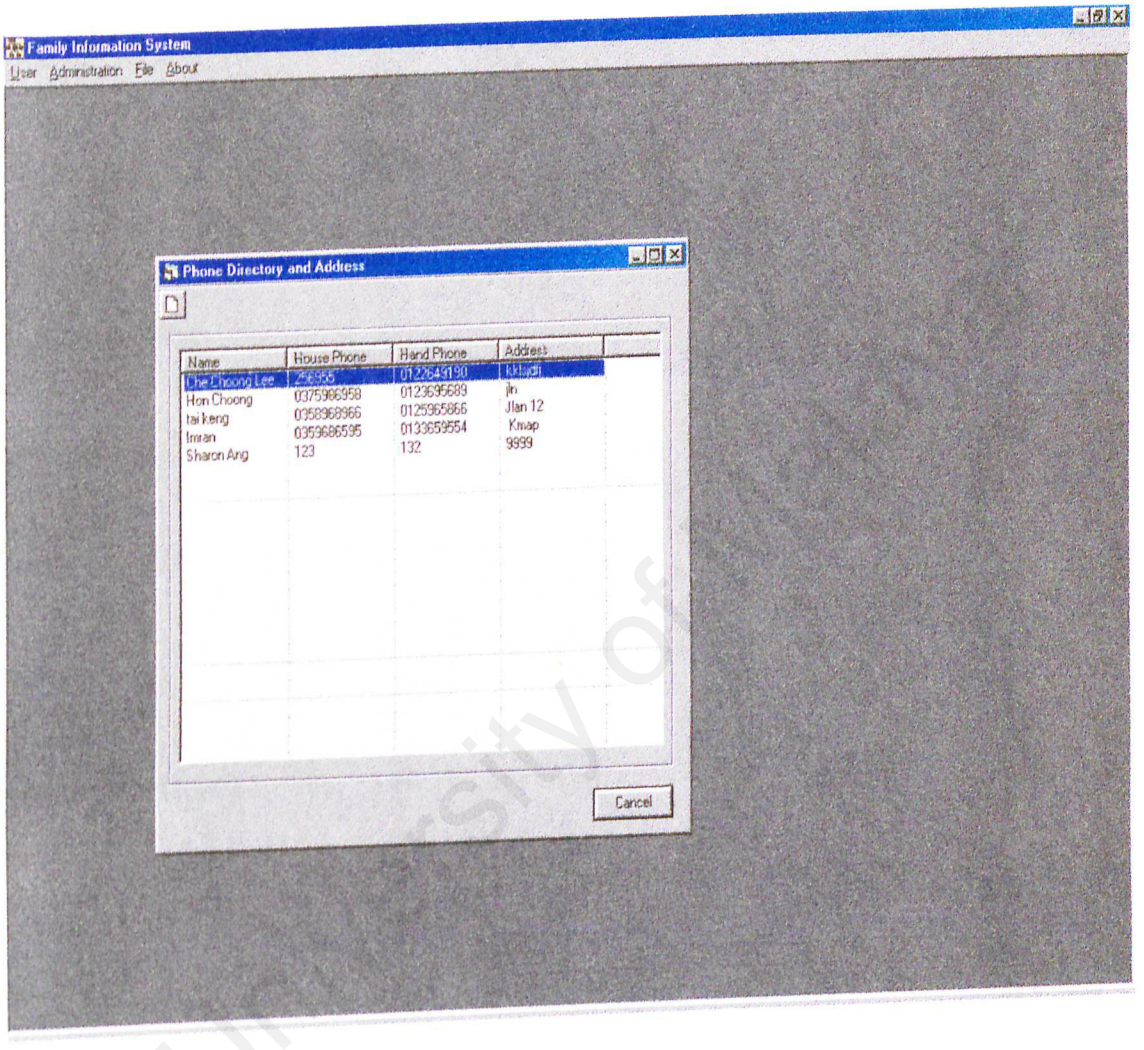
Example of the Save, Add, Delete, and New Operation for The FamilyIS



In this screen, user can add, save, create new form and delete existing data for the FamilyIS.

In this form, user can also add in their respective picture.

Example of Information Listing

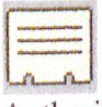


This screen will show all the telephone information of the concurrent user. User will be able to add in new record, delete existing record or even modify existing record.

Example of Command Button

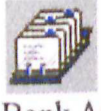

Change Password

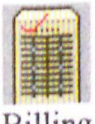

Authority Setup


Authority Info


Family Members

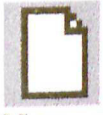

Assets


Bank Accounts


Billing


Logout

Exit Program



New

Open / Search

OTHER APPENDIX

BIBLIOGRAPHY

Internet Resources

1 **Ashok Ramachandran**, http://www.ashok.pair.com/

2 **Intergrow FMS**, http://www.hetco.com/faciliti.htm

3 **Indiana University,** http://www.iupui.edu

4 **Microsoft Cooperation,** http://www.microsoft.com

5 **Precision Inventory Management System,** http://education.software-directory.com

6 **Rhodes Facilities Management**, http://calendar.rhodes.edu/]

7 **Shadyside Hospital in Pittsburgh,** http://www.aperture.com

8 **University of Florida Health Science Center,** http://www.hsc.ufl.edu

9 **Webopedia,** http://webopedia.internet.com

Books And Publications

10 Connolly, Thomas M.; Begg, Carolyn E.; Strachan, Anne D.; "**Database System**", Harlow, England, 1996.

11 Fertuck, Len, **"System Analysis & Design"**, Wm. C.Brown Communications Inc, 1995.

12 Hawryszkiewycz, Igor; **"Introduction to System Analysis and Design (4th Edition)"** Prentice Hall Inc., 1997.

13 Kendall, Kenneth E. and Kendall, Julie E. **"System Analysis and Design 4th Edition"**, Prentice Hall International Inc. 1996.

14 Kroenke, David M.; **"Database Processing Fundamentals, Design, and Implementation 6th Edition"**, 1998.

15 Pfleeger, Shari Lawrence **"Software Engineering Theory and Practice"**, Prentice Hall, Inc. 1998.

16 Senn, James A.; **"Analysis and Design of Information System, 2nd Edition."** McGraw-Hill Publishing Company, 1989.

17 Sommerville, Ian; **"Software Engineering (5th Edition)"**, Addison Wesley USA, 1995.

18 Weinschenk, Susan; Jamar, Pamela; and Yeo, Sarah C.; **"GUI Design Essential for Windows® 95, Windows® 3.1, World Wide Web"**; John Wiley & Sons, Inc. 1997.

SISTEM PENGURUSAN INFORMASI KELUARGA

Nama Keluarga :
Nama responden :
Post :

Perhatian : *Segala maklumat yang diberikan dalam soal selidik ini adalah sulit dan hanya akan digunakan untuk tujuan analisis dan pembangunan Sistem Informasi Keluarga..*

Anda dibenarkan untuk memilih lebih daripada satu jawapan.

Maklumat dan Kemudahan

1. Apakah status keluarga anda?
   - ☐ **Keluarga Berinformasi**
   - ☐ **Keluarga Dalam Bandar**
   - ☐ **Keluarga Luar Bandar**

2. Apakah sistem pengendaliaan yang digunakan oleh komputer di rumah anda?
   - ☐ **Window 98**
   - ☐ **Window 95**
   - ☐ **Window yang lebih lama dari Win 95, sila nyatakan**_____
   - ☐ **Lain-lain, sila nyatakan**_____

3. Adakah anda berpuas hati dengan cara pengurusan informasi di keluarga anda?
   - ☐ **Tidak puas hati**
   - ☐ **Sedikit puas hati**
   - ☐ **Puas hati**
   - ☐ **Sangat puas hati**
   - ☐ **Tiada komen**

4. Pada pendapat anda, efektifkah sistem pengurusan informasi di rumah anda?
   - ☐ **Ya** ☐ **Tidak**

5. Adakah keluarga ini mempunyai sistem berkomputeran yang mengendalikan informasi keluarga?
   - ☐ **Ya** ☐ **Tidak**

6. Jika ya, apakah sistem pengendalian yang digunakan oleh sistem berkenaan?
   - ☐ **Window 98**
   - ☐ **Window 95**
   - ☐ **Window yang lebih lama dari Win 95, sila nyatakan**_____
   - ☐ **Lain-lain, sila nyatakan**_____

7. Nyataakan perisian yang digunakan oleh sistem tersebut.

_____

8. Sila berikan takrifan bagi informasi dalam konteks keluarga.
   - ☐ **Segala bentuk peralatan yang digunakan untuk membantu kehidupan harian.**
   - ☐ **Segala jenis maklumat peribadi.**
   - ☐ **Segala bentuk peralatan dan maklumat peribadi.**
   - ☐ **Lain-lain, sila nyatakan**_____

9. Siapakah yang menentukan penggunaan sesuatu informasi di rumah?
   - ☐ **Ibu dan Bapa**
   - ☐ **Anak**
   - ☐ **Penolong Rumah**
   - ☐ **Lain-lain, sila nyatakan**_____

10. Siapakah yang bertanggungjawab terhadap keselamatan informasi dan asset dalam rumah?
   - ☐ **Ibu dan Bapa**
   - ☐ **Anak**
   - ☐ **Penolong Rumah**
   - ☐ **Lain-lain, sila nyatakan**_____

11. Siapakah yang bertanggung jawab terhadap penjadualan aktiviti keluarga?
   - ☐ **Ibu dan Bapa**
   - ☐ **Anak**
   - ☐ **Penolong Rumah**
   - ☐ **Lain-lain, sila nyatakan**_____

12. Selain daripada IbuBapa dan anak, siapakah yang turut menggunakan informasi dan asset di rumah.
   - ☐ **Syarikat Insurans**
   - ☐ **Jabatan Kerajaan**
   - ☐ **Orang ramai selain ibu bapa dan penjaga**
   - ☐ **Lain-lain, sila nyatakan**_____

13. Apakah informasi dan asset yang digunakan oleh mereka?
    - ☐ **Nombor IC**
    - ☐ **Nombor Serial Asset**
    - ☐ **Sijil Kelahiran**
    - ☐ **Maklumat Bil-Bil rumah**
    - ☐ **Maklumat Telefon**
    - ☐ **Maklumat Peribadi**

Maklumat-maklumat yang ingin saya kumpulan untuk pangkalan data.

1. Senarai kemudahan rumah
2. Senarai penggunaan yang mungkin untuk setiap asset tersebut.
3. Senarai alatan dalam keluarga tesebut.
4. Susut nilai bagi setiap asset tersebut.

Questionnaire Analysis

Out of 5 questionnaires distributed to families, only one have gave full co-operation in replying my questionnaire. So, this analysis is done from the answer given by the particular family. So, it cannot be concluded that this analysis represents the management of information in the family.

Information

From the answer given by respondent, it can be concluded that the family is not using any type of computerised family information system.

Information in the family is manage in such a way that every information relating to the family is being stored manually. The particular person who is in charge of the family will also be in responsible to the safety of information like birth certificate and IC stored inside the house.

There are no other people (outsiders) who use the information in the family.

Assets

In doing inventory, the respondent do not have any specified ways of keeping records of all the assets in house. They do their inventory once a year, and there is only the person in charge of the house who is responsible of carrying out the inventory. Counting all the assets in the house, location by location, does inventory in the particular family, which include everything (CDs, licenses, brooms, chairs, tables, etc).